



НАЦИОНАЛЬНЫЙ
СТАНДАРТ
РОССИЙСКОЙ
ФЕДЕРАЦИИ

ГОСТ Р
53556.8 –
2013

Звуковое вещание цифровое

**КОДИРОВАНИЕ СИГНАЛОВ ЗВУКОВОГО ВЕЩАНИЯ С
СОКРАЩЕНИЕМ ИЗБЫТОЧНОСТИ ДЛЯ ПЕРЕДАЧИ ПО
ЦИФРОВЫМ КАНАЛАМ СВЯЗИ. ЧАСТЬ III
(MPEG-4 AUDIO)**

**Параметрическое кодирование высококачественных звуковых сигналов
(SSC)**

Издание официальное



Москва
Стандартинформ
2014

Предисловие

1 РАЗРАБОТАН Санкт-Петербургским филиалом Центрального научно-исследовательского института связи «Ленинградское отделение» (ФГУП ЛО ЦНИИС)

2 ВНЕСЕН Техническим комитетом по стандартизации № 480 «Связь»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 25 октября 2013 г. № 1212-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений международного стандарта ИСО/МЭК 14496–3:2009 «Информационные технологии. Кодирование аудиовизуальных объектов. Часть 3. Аудио» (ИСО/МЭК 14496–3:2009 Information technology - Coding of audio-visual objects - Part 3: Audio) [1]

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0–2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок – в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования – на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (gost.ru).

© Стандартинформ, 2013

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1	Область применения.....
2	Термины и определения.....
3	Символы и сокращения.....
3.1	Арифметические операторы
3.2	Операторы отношения
3.3	Мнемоники
3.4	Диапазоны
3.5	Численное представление.....
3.6	Определения
4	Полезные нагрузки для <i>SSC</i> аудио объектного типа
4.1	Конфигурация декодера (<i>SSCSpecificConfig</i>)
4.2	Полезная нагрузка потока битов <i>SSC</i>
5	Семантика.....
5.1	<i>SSCSpecificConfig</i>
5.2	Декодирование полезной нагрузки потока битов <i>SSC</i>
5.3	Индексация подфреймов
6	Процесс декодирования.....
6.1	Переходные процессы.....
6.2	Синусоиды.....
6.3	Шум.....
6.4	Параметрическое стерео.....
6.5	Ситуации запуска/остановки для декодирования.....
	Приложение А (обязательное) Комбинация инструмента <i>SBR</i> с параметрическим инструментом стерео.....
	Приложение Б (обязательное) Нормативные таблицы.....
	Приложение В (справочное) Темп и подача, масштабируемые в декодере
	Библиография.....

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Звуковое вещание цифровое

КОДИРОВАНИЕ СИГНАЛОВ ЗВУКОВОГО ВЕЩАНИЯ С
СОКРАЩЕНИЕМ ИЗБЫТОЧНОСТИ ДЛЯ ПЕРЕДАЧИ ПО ЦИФРОВЫМ
КАНАЛАМ СВЯЗИ.
ЧАСТЬ III (MPEG-4 AUDIO)Параметрическое кодирование высококачественных звуковых сигналов
(SSC)

Sound broadcasting digital.

Coding of signals of sound broadcasting with reduction of redundancy for transfer on digital
communication channels. A part III (MPEG-4 audio).

Technical description of parametric coding for high quality audio

Дата введения 2014-09-01

1 Область применения

Стандарт описывает схему параметрического кодирования для сжатия высококачественного аудио *MPEG-4 audio*. Краткое наименование - *SSC (Sinusoidal Coding)*. При скоростях передачи приблизительно 24 Кбит/с стерео и на частоте дискретизации 44,1 кГц схема кодирования *SSC* предлагает качество, которое необходимо для многих приложений.

SSC использует четыре различных инструмента, которые вместе параметризуют аудиосигнал. Эти инструменты состоят из моделирования переходного процесса, моделирования синусоид, моделирования шума и моделирования образа стерео. Одна из отличительных особенностей *SSC* состоит в том, что оно оказывает поддержку декодера для независимого темпа и масштабирования шага при почти любой дополнительной сложности.

Инструмент переходного процесса

Инструмент переходного процесса захватывает чрезвычайно динамичные события входного аудиосигнала. Эти события эффективно моделируются посредством ограниченного количества синусоид, которые формируются посредством огибающей.

Инструмент синусоид речевого выхода для инструментов *FA* и для дублирования *MP* с информацией о форме губ.

Инструмент синусоид захватывает детерминированные события входного аудиосигнала. Природа медленного изменения синусоидальных компонентов для типичных аудиосигналов используется, соединяя синусоиды по последовательным фреймам. Посредством дифференциального кодирования могут быть эффективно представлены параметры частоты, амплитуды и фазы.

Шумовой инструмент

Шумовой инструмент захватывает стохастические или недетерминированные события входного аудиосигнала. В декодере в качестве возбуждения используется генератор белого шума. Чтобы управлять временными и спектральными свойствами шума в аудиосигнале применяется временная и спектральная огибающая.

Инструмент параметрического кодирования стерео

Инструмент параметрического кодирования стерео в состоянии захватывать стереообраз входного аудиосигнала в ограниченном количестве параметров, требуя только небольшого служебного сигнала в пределах от нескольких Кбит/с для среднего качества до приблизительно 9 Кбит/с для более высокого качества. Вместе с монофоническим смешиванием входного сигнала стерео, сгенерированного параметрическим инструментом кодирования стерео, инструмент параметрического декодирования стерео в состоянии регенерировать стереосигнал. Это - универсальный инструмент, который может работать в комбинации с любым монофоническим кодером. В Приложении А настоящего стандарта приводится нормативное описание комбинации *HE-AAC* с инструментом параметрического кодирования стерео. *SSC* также может работать в двойном режиме моно. В этом случае инструмент параметрического кодирования стерео не используется. Инструмент параметрического стерео предназначен для низких скоростей передачи.

2 Термины и определения

2.1 фрейм: Основная единица, которая может декодироваться сама по себе (требуется информация заголовка файла для общих настроек декодера).

2.2 фильтр Лагеррра: Структура фильтра, используемого в анализе шумов и синтезе.

2.3 аудио фрейм: Содержит все данные, чтобы декодировать *SSC*-кодированный фрейм как автономный модуль (требуется информация заголовка файла для общих настроек декодера). Для аудио фреймов с *refresh_sinusoids* = %1 и *refresh_noise* = %1 полный фрейм всегда может быть восстановлен, иначе в случае произвольного доступа возможно, что части сигнала не могут быть восстановлены (например, синусоидальные продолжения, шум).

2.4 подфрейм: Мелкоструктурированность в пределах фрейма.

2.5 f_s : Частота дискретизации в герцах.

2.6 сегмент: Интервал выборок, которые могут быть синтезированы на основе параметров, которые соответствуют подфрейму. Размер сегмента $2 \cdot S$.

2.7 окно: Функция, которая используется, чтобы взвешивать синтезируемые выборки в пределах сегмента так, чтобы получить достоверный синтез.

2.8 LSF : Спектральная частота линии.

2.9 наложение и добление: Аддитивный метод объединения перекрывающихся интервалов во время синтеза сигнала.

2.10 процесс связывания: Метод отслеживания синусоидальных компонент в течение продолжительного времени.

2.11 рождение: Первый компонент синусоидальной дорожки.

2.12 продолжение: Компонент синусоидальной дорожки, который находится не в начале или конце дорожки.

2.13 смерть: Последний компонент синусоидальной дорожки.

2.14 SMR : Отношение сигнала к маскировке.

2.15 частичный: Синусоида ограниченной продолжительности.

2.16 IID : Межканальные различия в интенсивности.

2.17 IPD : Межканальная разность фаз.

2.18 OPD : Полная Разность фаз.

2.19 ICC : Межканальная когерентность.

3 Символы и сокращения

3.1 Арифметические операторы

$\lfloor x \rfloor$ Округление x к минус бесконечность

$\lceil x \rceil$ Округление x к плюс бесконечность.

mod Оператор модуля: $mod(x, y) = x - \frac{\lfloor x/y \rfloor \cdot y}{\lfloor x/y \rfloor}$. Определенный только для положительных значений x и y .

Γ Гамма функция распределения, определенная как $\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$

3.2 Операторы отношения

$x ? y : z$ Если x является истиной, тогда y , иначе z .

3.3 Мнемоники

Чтобы описать различные типы данных, используемые в кодированном потоке битов, определяются следующие мнемоники.

- uimsbf* Целое число без знака, старший значащий бит в начале.
- simsbf* Целое число со знаком, старший значащий бит в начале.
- bslbf* Левый бит потока битов в начале

3.4 Диапазоны

- [0, 10] Число в диапазоне от 0 до, и включая 10.
- [0, 10> Число в диапазоне от 0 до, но исключая 10.

3.5 Численное представление

- %X* Представление двоичного числа (например, %01111100).
- SX* Представление шестнадцатеричного числа (например, 7\$C).
- X* Числа без префикса используют представление десятичного числа (например, 124).

3.6 Определения

- S* Число выборок в подфрейме.
- L* Число выборок в сегменте; $L = 2 * S$.
- numQMFSlots* Число выборок поддиапазона *QMF* на элемент *ps_data* (). Для *SSC* этот параметр фиксируется в 24.

4 Полезные нагрузки для *SSC* аудио объектного типа

4.1 Конфигурация декодера (*SSCSpecificConfig*)

Синтаксис *SSCSpecificConfig* () смотри таблицу1.

Таблица 1 - Синтаксис *SSCSpecificConfig* ()

Синтаксис	Количество битов	Мнемоника
<pre> SSCSpecificConfig (channelConfiguration) { decoder_level update rate synthesis_method if (channelConfiguration != 1) { mode_ext if ((channelConfiguration == 2) && (mode_ext == 1)) { reserved } } } </pre>	<p>2</p> <p>4</p> <p>2</p> <p>2</p> <p>2</p>	<p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p>

4.2 Полезная нагрузка потока битов SSC

Полезная нагрузка потока битов SSC приведена в таблицях 2-14.

Таблица 2 - Синтаксис *ssc_audio_frame* ()

Синтаксис	Количество битов	Мнемоника
<pre> ssc_audio_frame () { ssc_audio_frame_header() ssc_audio_frame_data() } </pre>		

Таблица 3 - Синтаксис *ssc_audio_frame_header()*

Синтаксис	Количество битов	Мнемоника
<i>ssc_audio_frame_header()</i>		
{		
<i>refresh_sinusoids</i>	1	<i>uimsbf</i>
<i>refresh_sinusoids_next_frame</i>	1	<i>uimsbf</i>
<i>refresh_noise</i>	1	<i>uimsbf</i>
for (<i>ch</i> = 0; <i>ch</i> < <i>nrof_channels</i> ; <i>ch</i> ++)		
{		
<i>s_nrof_continuations</i> [0][<i>ch</i>]	Примечание	<i>uimsbf</i>
}		
<i>n_nrof_den</i>	5	<i>uimsbf</i>
<i>n_nrof_lsf</i>	Примечание	<i>uimsbf</i>
<i>freq_granularity</i>	2	<i>uimsbf</i>
<i>amp_granularity</i>	2	<i>uimsbf</i>
<i>phasejitter_present</i>	1	<i>uimsbf</i>
if (<i>phase_jitter_present</i> == 1)		
{		
<i>phasejitter_percentage</i>	2	<i>uimsbf</i>
<i>phasejitter_band</i>	2	<i>uimsbf</i>
}		
}		
Примечание: См. описание <i>s_nrof_continuations</i> и <i>n_nrof_lsf</i> в 5.2.		

Таблица 4 - Синтаксис *ssc_audio_frame_data* ()

Синтаксис	Количество битов	Мнемоника
<pre> <i>ssc_audio_frame_data</i>() { for (<i>sf</i> = 0; <i>sf</i> < <i>nrof_subframes</i>; <i>sf</i>++) { for (<i>ch</i> = 0; <i>ch</i> < <i>nrof_channels</i>; <i>ch</i>++) { <i>ssc_mono_subframe</i>(<i>sf</i>,<i>ch</i>) if ((<i>channelConfiguration</i> == 2) && (<i>mode_ext</i> == 1) && (<i>mod</i>(<i>sf</i>+1,4) == 0)) { <i>ps_data</i>() } } } } </pre>		

Таблица 5 - Синтаксис *ssc_mono_subframe* ()

Синтаксис	Количество битов	Мнемоника
<pre> <i>ssc_mono_subframe</i> (<i>sf</i>,<i>ch</i>) { <i>subframe_transients</i>(<i>sf</i>, <i>ch</i>) <i>subframe_sinusoids</i>(<i>sf</i>, <i>ch</i>) <i>subframe_noise</i>(<i>sf</i>, <i>ch</i>) } </pre>		

Таблица 6 - Синтаксис *subframe_transients* ()

Синтаксис	Количество битов	Мнемоника
<pre> subframe_transients (sf, ch) { t_transient_present[sf][ch] if (t_transient_present[sf][ch] == 1) { t_loc[sf][ch] if (t_type[sf][ch]==1) { t_b_par[sf][ch] t_chi_par[sf][ch] t_nrof_sin[sf][ch] t_nrof_sin[sf][ch]++ for (i = 0; i < t_nrof_sin[sf][ch]; i++) { t_freq[sf][ch][i] t_amp[sf][ch][i] t_phi[sf][ch][i] } } } } </pre>	<p>1</p> <p>Примечание</p> <p>2</p> <p>3</p> <p>3</p> <p>3</p> <p>9</p> <p>5</p> <p>5</p>	<p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>simsbf</i></p>
Примечание: См. описание <i>t_loc</i> в 5.2.		

Таблица 7 - Синтаксис *subframe_sinusoids()*

Синтаксис	Количество битов	Мнемоника
<pre> subframe_sinusoids(sf, ch) { n = 0; p=0; q=0; /* Continuations */ if (sf > 0) { noc = 0; while (tmp_cont[ch][noc] > 0) { noc++; } s_nrof_continuations[sf][ch] = noc; } if ((refresh_sinusoids == 1) && (sf == 0)) { for (i = 0; i < s_nrof_continuations[sf][ch]; i++, n++) { s_cont[sf][ch][n] = ssc_huff_dec(huff_scont,bs_codeword); s_freq_coarse[sf][ch][n] = ssc_huff_dec(huff_sfreqc,bs_codeword); s_freq_fine[sf][ch][n] s_amp_coarse[sf][ch][n] = ssc_huff_dec(huff_sampca,bs_codeword); s_amp_fine[sf][ch][n] s_phi[sf][ch][n] if (s_cont[sf][ch][n] > 0) { s_adpcm_grid[sf][ch][n] = ssc_huff_dec(huff_sgrid,bs_codeword); s_delta_cont_freq_pha[sf+1][ch][p] p++; } if (s_cont[sf][ch][n] > 1) { s_delta_cont_freq_pha[sf+2][ch][q] q++; } } } } </pre>	<p>2.5</p> <p>7..25</p> <p>0..3</p> <p>3..16</p> <p>0..3</p> <p>5</p> <p>3..7</p> <p>2</p> <p>2</p>	<p><i>bslbf</i></p> <p><i>bslbf</i></p> <p><i>simsbf</i></p> <p><i>bslbf</i></p> <p><i>simsbf</i></p> <p><i>simsbf</i></p> <p><i>bslbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p>

Продолжение таблицы 7		
Синтаксис	Количество битов	Мнемоника
<pre> else { for (i = 0; i < s_nrof_continuations[sf][ch]; i++, n++) { if (sf == 0) { s_cont[sf][ch][n] = ssc_huff_dec(huff_scont, bs_codeword); } else { s_cont[sf][ch][n] = tmp_cont[ch][n] - 1; } if (s_cont[sf][ch][n] > 0) { p++; } if (s_cont[sf][ch][n] > 1) { if ((refresh_sinusoids_next_frame == 0) (nrof_subframes - sf > 2)) { s_delta_cont_freq pha[sf+2][ch][q] } q++; } s_delta_cont_amp[sf][ch][n] = ssc_huff_dec(huff_sampr[amp_granularity], bs_codeword); } } /* Births */ s_nrof_births[sf][ch] = ssc_huff_dec(huff_nrofbirths, bs_codeword); if (s_nrof_births[sf][ch] > 0) { s_cont[sf][ch][n] = ssc_huff_dec(huff_scont, bs_codeword); s_freq_coarse[sf][ch][n] = ssc_huff_dec(huff_sfreqba, bs_codeword); </pre>	<p>2..5</p> <p>2</p> <p>1..15</p> <p>3..15</p> <p>2..5</p> <p>7..21</p>	<p><i>bslbf</i></p> <p><i>uimsbf</i></p> <p><i>bslbf</i></p> <p><i>bslbf</i></p> <p><i>bslbf</i></p>

Продолжение таблицы 7		
Синтаксис	Количество битов	Мнемоника
<i>s_freq_fine[sf][ch][n]</i>	0..3	<i>simsbf</i>
<i>s_amp_coarse[sf][ch][n] =</i> <i>ssc_huff_dec(huff_sampba,bs_codeword);</i>	3..15	<i>bslbf</i>
<i>s_amp_fine[sf][ch][n]</i>	0..3	<i>simsbf</i>
<i>s_phi[sf][ch][n]</i>	5	<i>simsbf</i>
<i>if (s_cont[sf][ch][n] > 0)</i> <i>{</i> <i> if ((refresh_sinusoids_next_frame == 0) (nrof_subframes-sf ></i> <i>1))</i> <i> {</i> <i> s_delta_cont_freq_pha[sf+1][ch][p]</i>	2	<i>uimsbf</i>
<i> }</i> <i> p++;</i> <i>}</i> <i>if (s_cont[sf][ch][n] > 1)</i> <i>{</i> <i> if ((refresh_sinusoids_next_frame == 0) (nrof_subframes-sf > 2))</i> <i> {</i> <i> s_delta_cont_freq_pha[sf+2][ch][q]</i>	2	<i>uimsbf</i>
<i> }</i> <i> q++;</i> <i>}</i> <i>n++;</i> <i>for (i = 1; i < s_nrof_births[sf][ch]; i++, n++)</i> <i>{</i> <i> s_cont[sf][ch][n] = ssc_huff_dec(huff_scont,bs_codeword);</i>	2..5	<i>bslbf</i>
<i> s_delta_birth_freq_coarse[sf][ch][n] =</i> <i> ssc_huff_dec(huff_sfreqbr,bs_codeword);</i>	5..23	<i>bslbf</i>
<i> s_delta_birth_freq_fine[sf][ch][n]</i>	0..3	<i>simsbf</i>
<i> s_delta_birth_amp_coarse[sf][ch][n] =</i> <i> ssc_huff_dec(huff_sampbr,bs_codeword);</i>	2..21	<i>bslbf</i>
<i> s_delta_birth_amp_fine[sf][ch][n]</i>	0..3	<i>simsbf</i>
<i> s_phi[sf][ch][n]</i>	5	<i>simsbf</i>

Таблица 8 - Синтаксис *subframe_noise()*

Синтаксис	Количество битов	Мнемоника
<pre> subframe_noise (sf, ch) { if ((refresh_noise == 1) && (sf == 0)) { n_laguerre[ch] n_laguerre_granularity[sf][ch] for (i = 0; i < n_nrof_den; i++) { n_lar_den_coarse[sf][ch][i] = ssc_huff_dec(huff_nlag,bs_codeword); If (n_laguerre_granularity[sf][ch]==1) { n_lar_den_fine[sf][ch][i] } } n_gain[sf][ch] n_lsf[sf][ch][0] = ssc_huff_dec(huff_nlsf,bs_codeword); for (i = 1; i < n_nrof_lsf; i++) { n_delta_lsf[sf][ch][i] = ssc_huff_dec(huff_nlsf,bs_codeword); } } else { if (mod(sf,2) == 0) { n_laguerre_granularity[sf][ch] for (i = 0; i < n_nrof_den; i++) { n_delta_lar_den_coarse[sf][ch][i] = ssc_huff_dec(huff_nlag,bs_codeword); if(n_laguerre_granularity[sf][ch]==1) </pre>	<p>2</p> <p>1</p> <p>1..18</p> <p>2</p> <p>7</p> <p>2..9</p> <p>2..9</p> <p>1</p> <p>1..18</p>	<p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>bslbf</i></p> <p><i>simsbf</i></p> <p><i>uimsbf</i></p> <p><i>bslbf</i></p> <p><i>bslbf</i></p> <p><i>uimsbf</i></p> <p><i>bslbf</i></p>

Окончание таблицы 8		
Синтаксис	Количество битов	Мнемоника
<pre> { n_delta_lar_den_fine[sf][ch][i] } } } if (mod(sf,4) == 0) { n_delta_gain[sf][ch] = ssc_huff_dec(huff_ngain,bs_codeword); if (n_overlap_lsf == 1) { for (i = n_nrof_overlap_lsf; i < n_nrof_lsf; i++) { n_delta_lsf[sf][ch][i] = ssc_huff_dec(huff_nlsf,bs_codeword); } } else { n_lsf[sf][ch][0] = ssc_huff_dec(huff_nlsf,bs_codeword); for (i = 1; i < n_nr_of_lsf; i++) { n_delta_lsf[sf][ch][i] = ssc_huff_dec(huff_nlsf,bs_codeword); } } } } } </pre>	<p>2</p> <p>1..12</p> <p>1</p> <p>2..9</p> <p>2..9</p> <p>2..9</p>	<p><i>simsbf</i></p> <p><i>bslbf</i></p> <p><i>uimsbf</i></p> <p><i>bslbf</i></p> <p><i>bslbf</i></p> <p><i>bslbf</i></p>

Таблица 9 — Синтаксис *ps_data()*

Синтаксис	Количество битов	Мнемоника
<i>ps_data()</i>		
{		
<i>if(enable_ps_header) {</i>	1	<i>uimsbf</i>
<i>if(enable_iid) {</i>	1	<i>uimsbf</i>
<i>iid_mode</i>	3	<i>uimsbf</i>
<i>nr_iid_par = nr_iid_par_tab[iid_mode]</i>		
<i>nr_ipdopd_par =</i>		
<i>nr_ipdopd_par_tab[iid_mode]</i>		
}		
<i>if(enable_icc) {</i>	1	<i>uimsbf</i>
<i>icc_mode</i>	3	<i>uimsbf</i>
<i>nr_icc_par = nr_icc_par_tab[icc_mode]</i>		
}		
<i>enable_ext</i>	1	<i>uimsbf</i>
}		
<i>frame_class</i>	1	<i>uimsbf</i>
<i>num_env_idx</i>	2	<i>uimsbf</i>
<i>num_env = num_env_tab[frame_class][num_env_idx]</i>		
<i>if(frame_class) {</i>		
<i>for (e=0 ; e<num_env ; e++) {</i>		
<i>border_position[e]</i>	5	<i>uimsbf</i>
}		
}		
<i>for (e=0 ; e<num_env ; e++) {</i>		
<i>if(enable_iid) {</i>		
<i>iid_dt[e]</i>		
<i>iid_data()</i>	1	<i>uimsbf</i>
}		
}		
<i>for (e=0 ; e<num_env ; e++) {</i>		
<i>if(enable_icc) {</i>		

Окончание таблицы 9		
Синтаксис	Количество битов	Мнемоника
<pre> <i>icc_dt[e]</i> <i>icc_data()</i> } } <i>if (enable_ext) {</i> <i>cnt = ps_extension_size</i> <i>if (cnt == 15)</i> <i>cnt += esc_count</i> <i>num_bits_left = 8 * cnt</i> <i>while (num_bits_left > 7) {</i> <i>ps_extension_id</i> <i>num_bits_left -= 2</i> <i>ps_extension(ps_extension_id, num_bits_left)</i> } <i>fill_bits</i> } } </pre>	<p>1</p> <p>4</p> <p>8</p> <p>2</p> <p><i>num_bits_left</i></p>	<p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p>

Таблица 10 — Синтаксис *ps_extension()*

Синтаксис	Количество битов	Мнемоника
<pre> <i>ps_extension(ps_extension_id, num_bits_left)</i> { <i>if (ps_extension_id == 0) {</i> <i>if (enable_ipdopd) {</i> <i>for (e=0 ; e<num_env ; e++) {</i> <i>ipd_dt[e]</i> <i>ipd_data()</i> <i>opd_dt[e]</i> <i>opd_data()</i> <i>num_bits_left -= ipd_bits + opd_bits + 2</i> } } } </pre>	<p>1</p> <p>1</p> <p>1</p>	<p><i>uimsbf</i></p> <p><i>uimsbf</i></p> <p><i>uimsbf</i></p>

Окончание таблицы 10		
Синтаксис	Количество битов	Мнемоника
<pre> } reserved_ps num_bits_left -= 2 } } </pre>	1	<i>uimsbf</i>

Таблица 11 — Синтаксис *iid_data()*

Синтаксис	Количество битов	Мнемоника
<pre> iid_data() if (iid_dt[e]) { for (b=0 ; b<nr_iid_par; b++) { iid_par_dt[e][b] = ssc_huff_dec(huff_iid_dt[iid_quant], bs_codeword); } } else { for (b=0 ; b<nr_iid_par; b++) { iid_par_df[e][b] = ssc_huff_dec(huff_iid_df[iid_quant], bs_codeword); } } } </pre>	<p>1...20</p> <p>1...18</p>	

Таблица 12 — Синтаксис *icc_data()*

Синтаксис	Количество битов	Мнемоника
<pre> icc_data() { if (icc_dt[e]) { for (b=0 ; b<nr_icc_par; b++) { icc_par_dt[e][b] = ssc_huff_dec(huff_icc_dt,bs_codeword); } } else { for (b=0 ; b<nr_icc_par; b++) { icc_par_df[e][b] = ssc_huff_dec(huff_icc_df,bs_codeword); } } } </pre>	1...14	<i>bslbf</i>
	1...13	<i>bslbf</i>

Таблица 13 — Синтаксис *ipd_data()*

Синтаксис	Количество битов	Мнемоника
<pre> ipd_data() { if (ipd_dt[e]) { for (b=0 ; b<nr_ipdopd_par; b++) { ipd_par_dt[e][b] = ssc_huff_dec(huff_ipd_dt, bs_codeword); } } else { for (b=0 ; b<nr_ipdopd_par; b++) { ipd_par_df[e][b] = ssc_huff_dec(huff_ipd_df,bs_codeword); } } } </pre>	1...5	<i>bslbf</i>
	1...4	<i>bslbf</i>

Таблица 14 — Синтаксис *opd_data()*

Синтаксис	Количество битов	Мнемоника
<pre> opd_data() { if (opd_dt[e]) { for (b=0 ; b<nr_ipdopd_par; b++) { opd_par_dt[e][b] = ssc_huff_dec(huff_opd_dt,bs_codeword); } } else { for (b=0 ; b<nr_ipdopd_par; b++) { opd_par_df[e][b] = ssc_huff_dec(huff_opd_df,bs_codeword); } } } </pre>	1...5	<i>bslbf</i>
	1...5	<i>bslbf</i>

5 Семантика

5.1 SSCSSpecificConfig

decoder_level – пределы сложности для настроек декодера. Декодер, который поддерживает определенный уровень сложности, не в состоянии декодировать поток битов, который кодируется согласно более высокому уровню сложности. Этот декодер в состоянии декодировать поток битов, который кодируется согласно более низкому уровню сложности (таблица 15).

Таблица 15 - Уровень декодера

<i>decoder_level</i>	Уровень сложности	<i>max_nrof_sins</i>	<i>max_nrof_den</i>	Число битов для <i>s_nrof_continuation</i>	Число битов для <i>n_nrof_lsf</i>
00	Зарезервировано	<i>Na</i>	<i>Na</i>	<i>Na</i>	<i>Na</i>
01	Носитель	60	24	6	4
10	Зарезервировано	<i>Na</i>	<i>Na</i>	<i>Na</i>	<i>Na</i>
11	Зарезервировано	<i>Na</i>	<i>Na</i>	<i>Na</i>	<i>Na</i>

ГОСТ Р 53556.8 - 2013

max_nrof_sinusoids - максимальное количество синусоид, которое допускается

max_nrof_den - максимальное значение для *n_nrof_den*.

update_rate - четыре бита, указывающие размер подфрейма *S*.

Таблица 16 показывает соотношение между *update_rate* и размером подфрейма *S* в выборках.

Таблица 16 - Уровень обновления

<i>update_rate</i>	<i>S</i>	<i>update_rate</i>	<i>S</i>
0000	Зарезервировано	1000	Зарезервировано
0001	Зарезервировано	1001	Зарезервировано
0010	Зарезервировано	1010	Зарезервировано
0011	Зарезервировано	1011	Зарезервировано
0100	384	1100	Зарезервировано
0101	Зарезервировано	1101	Зарезервировано
0110	Зарезервировано	1110	Зарезервировано
0111	Зарезервировано	1111	Зарезервировано

synthesis_method - два бита, предоставляющие информацию о предпочтительном синтезе для специально закодированной программы (таблица 17).

Таблица 17 - Метод синтеза

<i>Synthesis_method</i>	Оптимальный синтез
00	Наложение и добавление
01	Зарезервировано
10	Зарезервировано
11	Зарезервировано

mode_ext - в сочетании с *channelConfiguration* биты *mode_ext* обеспечивают полную конфигурацию канала. Число битов зависит от *channelConfiguration* (таблица 18).

Таблица 18 - Конфигурация канала

<i>channelConfiguration</i>	Количество битов для <i>mode_ext</i>	<i>nrof_channels</i>
1	0	1
2	2	Согласно <i>mode_ext</i>
0, 3 ... 15	<i>Na</i>	<i>Na</i>

Для *channelConfiguration* == 2, применяется таблица 19:

Таблица 19 - Конфигурация канала в случае, если *channelConfiguration* == 2

<i>mode_ext</i>	Полная конфигурация канала	<i>nrof_channels</i>
00	Двойной моно (<i>ch0=left, ch1=right</i>)	2
01	Параметрическое Стерео	1
10	Зарезервировано	<i>Na</i>
11	Зарезервировано	<i>Na</i>

Зарезервировано - два зарезервированных бита; должно быть установлено в %0.

5.2 Декодирование полезной нагрузки потока битов SSC

ssc_audio_frame () - синтаксический элемент, который содержит единственный фрейм SSC

ssc_audio_frame_header () - синтаксический элемент, который содержит данные заголовка для единственного фрейма SSC

ssc_audio_frame_data () - синтаксический элемент, который содержит данные для единственного фрейма SSC

ssc_huff_dec () - процедура декодирования Хаффмана.

refresh_sinusoids - один бит, указывающий, как кодируются синусоидальные продолжения первого подфрейма во фрейме. Если этот бит равняется %0, данные продолжающейся дорожки дифференцированно кодируются относительно последнего подфрейма предыдущего фрейма. Если этот бит равняется %1, данные продолжающейся дорожки в первом подфрейме фрейма кодируются как абсолютные значения.

refresh_sinusoids_next_frame - один бит, обеспечивающий дополнительный предпросмотр фрейма для декодирования ADPCM синусоидальных параметров. Если этот бит устанавливается в %1, следующий фрейм является фреймом обновления. В этом случае бит *refresh_sinusoids* должен быть установлен в %1 в следующем фрейме. Если этот бит устанавливается в %0, следующий фрейм не является фреймом обновления.

refresh_noise - один бит, указывающий, как кодируются шумовые параметры первого подфрейма во фрейме. Если этот бит равняется %0, шумовые параметры дифференцированно кодируются относительно последнего подфрейма предыдущего фрейма. Если этот бит равняется %1, шумовые параметры в первом подфрейме фрейма кодируются как абсолютные значения.

s_nrof_continuations [*sf*] [*ch*] - для подфрейма *sf* и канала *ch*, это значение пред-

ставляет число продолжений. В случае $sf == 0$ значение $s_nrof_continuations$ обеспечивается в потоке битов. Для остальных значений sf значение $s_nrof_continuations$ получается неявно. Число битов, требуемых для $s_nrof_continuations [0] [ch]$, зависит от максимального количества разрешенных синусоид, которое зависит от сложности декодера, указанной в $decoder_level$.

n_nrof_den - число коэффициентов LAR знаменателя фильтра FIR для генерации шума.

n_nrof_lsf - число коэффициентов LSF , используемых для генерации огибающей для генерации шума. Число битов, требующихся для n_nrof_lsf , зависит от сложности декодера, указанной $decoder_level$.

$freq_granularity$ - гранулярность дифференцированно или абсолютно кодированных частотных параметров, используемых в $subframe_sinusoids ()$. Этот параметр определяет число битов, которые будут считаны из тонкоструктурной части параметров частоты.

$amp_granularity$ - гранулярность дифференцированно или абсолютно кодированных амплитудных параметров, используемых в $subframe_sinusoids ()$. Этот параметр определяет таблицу Хаффмана, которая будет использоваться, или число битов, которые будут считаны из тонкоструктурной части амплитудных параметров.

$phase_jitter_present$ - один бит, чтобы указать на присутствие параметров дрожания фазы. Если этот бит равняется %0, никакого дрожания фазы нет. Если этот бит равняется %1, дрожание фазы присутствует.

$phase_jitter_percentage$ - это двухбитовое целое число без знака, указывающее процент расстояния. Полное расстояние равняется половине шага квантования. Максимальное дрожание, применяемое к частотным компонентам

$$\max_jitter = 2^{freq_granularity} \cdot \frac{phase_jitter_percentage + 1}{2^2}$$

$phase_jitter_band$ - два бита, указывающие уровень представления частоты, из которого должно быть применено дрожание фазы, таблица 20.

Таблица 20 - Полоса дрожания фазы, выраженная в уровнях представления

$phase_jitter_band$	Уровень представления частоты $f_{jitter.min}$
00	0
01	800
10	1600
11	2400

nrof_subframes - число подфреймов в одном фрейме. Это значение фиксируется в 8.

ssc_mono_subframe () - синтаксический элемент, который содержит данные для одного подфрейма SSC.

ps_data () - синтаксический элемент, который содержит параметрические данные стерео.

subframe_transients () - синтаксический элемент, который содержит переходные данные для одного подфрейма SSC.

subframe_sinusoids () - синтаксический элемент, который содержит данные синусоид для одного подфрейма SSC.

subframe_noise () - синтаксический элемент, который содержит данные о шуме для одного подфрейма SSC.

t_transient_present [*sf*] [*ch*] - один бит, указывающий, присутствует ли переходное состояние в подфрейме *sf*, канала *ch*. Если *t_transient_present* [*sf*] [*ch*] = %1, переходное состояние присутствует. Если *t_transient_present* [*sf*] [*ch*] = %0, переходное состояние отсутствует.

t_loc [*sf*] [*ch*] - указание расположения переходного состояния в подфрейме *sf* канала *ch*, выраженное в числе выборок от начала подфрейма. Допустимый диапазон для *t_loc* равен [0, S]. Число битов, которое используется, чтобы представить *t_loc*, вычисляется согласно $\lceil \log_2(S) \rceil$, где *S* представляет размер подфрейма в выборках.

t_type [*sf*] [*ch*] - два бита для указания типа переходного состояния в подфрейме *sf* канала *ch* (таблица 21).

Таблица 21 – Типы переходного состояния

<i>t_type</i>	Тип
00	Шаг
01	Мейкснера (<i>Meixner</i>)
10	Зарезервировано
11	Зарезервировано

t_b_par [*sf*] [*ch*] - для переходного состояния типа Мейкснера в подфрейме *sf* канала *ch*, эти 3 бита содержат значение для атаки огибающей перехода, обозначенное как '*b*-параметр'. Допустимые значения для *t_b_par* [0, 1, 2, 3]. Остальные значения резервируются. Значение *b* вычисляется как $b = t_b_par + 2$.

t_chi_par [*sf*] [*ch*] - для переходного состояния типа Мейкснера в подфрейме *sf*

канала ch , эти 3 бита хранят значение для спада огибающей переходного состояния, обозначенное как ' ξ - параметр'. Допустимые значения для t_chi_par равны [0, 1, 2, 3]. Остающиеся значения резервируются. Значения ξ сведены в таблицу 22.

Таблица 22 - Квантованные значения для b и ξ

ξ		t_b_par			
		0	1	2	3
t_chi_par	0	0,9688	0,9685	0,9683	0,9681
	1	0,9763	0,9756	0,9750	0,9744
	2	0,9839	0,9827	0,9817	0,9807
	3	0,9914	0,9898	0,9884	0,9870

$t_nrof_sin [sf] [ch]$ - для переходного состояния типа Мейкснера в подфрейме sf канала ch , эти 3 бита представляют число синусоид, которые присутствуют под огибающей. Число синусоид под огибающей Мейкснера равно значению в потоке плюс один.

$t_freq [sf] [ch] [i]$ - для переходного состояния типа Мейкснера в подфрейме sf канала ch эти биты представляют частоту в радианах i -ой синусоиды под огибающей переходного состояния.

$$if[i] = \frac{2 \cdot 10^{\frac{t_freq[sf][ch][i] - 1}{11.4}}}{t_s \cdot 0.00437},$$

где t_f представляет деквантизованную абсолютную частоту в радианах.

$t_amp [sf] [ch] [i]$ - для переходного состояния типа Мейкснера в подфрейме sf канала ch эти биты представляют амплитуду i -ой синусоиды под огибающей перехода.

$$ta_q[i] = ta_b^{24 \cdot amp[sf][ch][i]},$$

где ta_b представляет логарифмическую базу квантования, $ta_b = 1885,1$. ta_b представляет деквантованную абсолютную амплитуду.

$t_phi [sf] [ch] [i]$ - для переходного состояния типа Мейкснера в подфрейме sf канала ch эти биты представляют фазу i -ой синусоиды под огибающей переходного состояния. Декодированное значение преобразовывается в значение фазы в радианах в диапазоне $[-\pi, \pi>$ и определяется для начала переходного состояния.

$$tp_q[i] = 2^4 p_e \cdot \mathcal{U}_{phi}[sf][ch][i],$$

где tp_e представляет абсолютную ошибку фазы ($tp_e = \frac{\rho}{32}$), и tp_q представляет де-

квантизованную абсолютную фазу (в радианах). Допустимый диапазон для t_phi равен [-16, 15].

noc - локальная переменная, которая считает число продолжений в предыдущем подфрейме.

$tmp_cont [ch] [noc]$ - локальный массив, который содержит копию s_cont -параметров предыдущего подфрейма, необходимую для того, чтобы правильно проанализировать поток (извлечь число продолжений и отслеживать сколько подфреймов должна продолжаться синусоидальная дорожка в текущем фрейме).

$s_cont [sf] [ch] [n]$ - для подфрейма sf и канала ch , эта величина указывает, сколько компонент подфреймов n будет продолжаться в текущем фрейме (если компонент будет продолжаться также в следующем фрейме к числу подкадров, которые он продолжается в текущем фрейме, должна быть добавлена единица. Если значение равно 0, это указывает на прекращение компонента n в подкадре sf , что называют смертью). Допустимый диапазон для s_cont составляет [0, 9].

$s_freq_coarse [sf] [ch] [n]$ - для подфрейма sf и канала ch это значение представляет грубый параметр частоты n -ой синусоиды.

$s_freq_fine [sf] [ch] [n]$ - для подфрейма sf и канала ch это целое число со знаком представляет более высокий уровень детализации дополнительно к грубому параметру частоты. Число битов, которые будут считаны достигает $(3 - freq_granularity)$. Уровень представления частоты f_{el} является суммой грубой частоты, точной частоты, масштабируемой по сетке гранулярности.

$$f_{el}[n] = s_freq_coarse[sf][ch][n] + s_freq_fine[sf][ch][n] \cdot 2^{freq_granularity}$$

Дрожание фазы применяется только в комбинации с масштабированием шага и темпа. Если $phase_jitter_present == \%1$ и $f_{el} > f_{jitter,min}$, параметр дрожания фазы будет

$$f_{jitter} \cdot \left(\max_jitter \cdot (2x - 1) + 0,5 \right)$$

где x содержит случайное число, равномерно распределенное между 0 и 1, сгенерированное для каждого параметра частоты в подфрейме, соответствующая вышеуказанным требованиям. Декодированное значение преобразовывается в деквантованное абсолютное значение частоты f_q в радианах, используя следующее уравнение:

$$f_q[n] = \frac{2p}{f_s} \frac{10^{\frac{f_{el}[n]}{91,241,4} - 1}}{0,00437}$$

$s_amp_coarse [sf] [ch] [n]$ - для подфрейма sf и канала ch это значение представляет грубый амплитудный параметр n -ой синусоиды.

$s_amp_fine [sf] [ch] [n]$ - Для подфрейма sf и канала ch этот параметр представляет

более высокий уровень детализации дополнительно к грубому амплитудному параметру. Число битов, которые будут считаны, насчитывает до $(3 - amp_granularity)$. Уровень представления амплитуды sa_n является суммой грубой амплитуды, точной амплитуды, масштабируемой по сетке гранулярности

$$sa_n[n] = s_amp_coarse[sf][ch][n] + s_amp_fine[sf][ch][n] \cdot 2^{amp_granularity}$$

Декодированное значение преобразовывается в деквантованное линейное значение амплитуды sa_q в диапазоне $[1, 2^{-15}]$ в соответствии с

$$sa_q[n] = sa_b^{24a_n[n]},$$

где $sa_b = 1,0218$ является базой логарифмического квантования. Его значение соответствует максимальной ошибке 0,1875 дБ.

$s_phi[sf][ch][n]$ - для подфрейма sf и канала ch это представляет фазовый параметр n -ой синусоиды. Это значение преобразовывается в значение фазы в радианах в диапазоне $[-\pi, \pi]$ в соответствии с

$$sp_q[n] = 2\psi_p \cdot \psi_phi[sf][ch][n],$$

где sp_p представляет ошибку абсолютной фазы ($sp_p \frac{p}{32}$), а sp_q представляет деквантизованную абсолютную фазу (в радианах). Допустимый диапазон для s_phi равен $[-16, 15]$.

$s_adpcm_grid[sf][ch][n]$ - для подфрейма sf и канала ch это значение представляет начальный индекс в таблице 35 как использующийся в декодере ADPCM для n -ой синусоиды. Эта таблица используется, чтобы декодировать информацию о синусоидах.

$s_delta_cont_freq_pha[sf][ch][n]$ - для подфрейма sf и канала ch это значение представляет уровни представления для n -ой синусоиды, которые служат вводом в декодер ADPCM. Чтобы компенсировать эту задержку декодера на 2 подфрейма, уровни представления передаются заранее на 2 подфрейма. В синтаксисе потока битов будущие уровни представления обозначаются индексами $sf+1$ и $sf+2$, указывающими уровни представления двух последующих подфреймов, соответственно. В случае, если $sf+1$ или $sf+2$ превышает $nrof_subframes$, уровень представления присваивается следующему фрейму. В этом случае новое число подфреймов в следующем фрейме будет $(sf+1) - nrof_subframes$ или $(sf+2) - nrof_subframes$ соответственно.

$s_delta_cont_amp[sf][ch][n]$ - для подфрейма sf и канала ch это представляет дифференциальный амплитудный параметр n -ой синусоиды. Это значение преобразовывается в линейное амплитудное значение в диапазоне $[1, 2^{-15}]$ в соответствии с

$$sa_n[n] = sa_{n,pf} + s_delta_cont_amp[sf][ch][n],$$

где sa_n представляет уровень представления амплитуды, а $sa_{n,pst}$ представляет уровень представления амплитуды в предыдущем подфрейме. Для деквантизации sa_n в sa_q смотрите s_amp_fine . В случае, если амплитудная гранулярность, $amp_granularity$ текущего фрейма отличается от гранулярности предыдущего фрейма, перед применением дифференциально кодированных значений гранулярность значения предыдущего фрейма преобразовывается в гранулярность текущего фрейма согласно

$$sa_{n,pst} = 2^{amp_granularity} \frac{sa'_{n,pst}}{2^{amp_granularity}} + 0,5 \frac{b}{B}$$

где $sa'_{n,pst}$ представляет уровень представления амплитуды предыдущего подфрейма, и $amp_granularity$ представляет гранулярность текущего подфрейма.

$s_nrof_births [sf] [ch]$ - Для подфрейма sf и канала ch это значение представляет число рождений. Разрешенный диапазон $[0, max_nrof_sinusoids - s_nrof_continuations [sf] [ch]]$.

$s_delta_birth_freq_coarse [sf] [ch] [n]$ - для подфрейма sf и канала ch это значение представляет дифференциальный, грубый параметр частоты n -ой синусоиды.

$s_delta_birth_freq_fine [sf] [ch] [n]$ - для подфрейма sf и канала ch это представляет более высокий уровень детализации в сравнении с грубым дифференциальным параметром частоты. Число битов, которые будут считаны, равно $(3 - freq_granularity)$. Уровень представления дельта-частоты df_n будет

$$df_n[n] = s_delta_birth_freq_coarse[sf][ch][n] + s_delta_birth_freq_fine[sf][ch][n] \cdot 2^{freq_granularity}$$

Декодированное значение n -ой синусоиды преобразовывается в значение частоты в герцах, используя уровень представления частоты предыдущего рождения f_n подкадра sf ($(n-1)$ -ая синусоида)

$$f_n[n] = f_n[n-1] + df_n[n],$$

где f_n представляет уровень представления частоты. Модификация f_n из-за дрожания фазы использует те же самые правила, как установленные под s_freq_fine . О деквантизации f_n в f_q смотрите также s_freq_fine .

$s_delta_birth_amp_coarse [sf] [ch] [n]$ - для подфрейма sf и канала ch это представляет дифференциальный, грубый амплитудный параметр n -ой синусоиды.

$s_delta_birth_amp_fine [sf] [ch] [n]$ - для подфрейма sf и канала ch это представляет более высокий уровень детализации грубого амплитудного параметра. Число битов, которые будут считаны, составляет $(3 - amp_granularity)$. Уровень дельта-

представления амплитуды sda_{rl}

$$sda_{rl}[n] = s_delta_birth_amp_coarse[sf][ch][n] + s_delta_birth_amp_fine[sf][ch][n] \cdot 2^{n_granularity}$$

Декодированное значение для n -ой синусоиды преобразовывается в линейное амплитудное значение, используя уровень представления амплитуды предыдущего рождения sa_{rl} ($(n-1)$ -ая синусоида):

$$sa_{rl}[n] = sa_{rl}[n-1] + sd_{rl}[n],$$

где sa_{rl} представляет уровень представления амплитуды.

$n_laguerre[ch]$ - λ коэффициент фильтра Лагерра для синтеза шума, см. таблицу 23.

Таблица 23 - Возможные значения для λ

$n_laguerre$	λ
00	0
01	0,5
10	0,7
11	Зарезервировано

$n_laguerre_granularity[sf][ch]$ - 1 бит, обозначающий точность квантования коэффициентов Лагерра.

$n_lar_den_coarse[sf][ch][i]$ - для подфрейма sf и канала ch это представляет номер коэффициента LAR знаменателя i .

$n_lar_den_fine[sf][ch][i]$ - для подфрейма sf и канала ch это представляет более высокий уровень детализации относительно грубого параметра коэффициента LAR знаменателя. Уровень представления $nlar_{rl}$ является суммой грубого LAR знаменателя и точного LAR знаменателя:

$$nlar_{rl}[i] = n_lar_den_coarse[sf][ch][i] + n_lar_den_fine[sf][ch][i]$$

и преобразовывается в коэффициент LAR согласно:

$$nlar_q[i] = nlar_{rl}[i] * \Delta_{LAR}$$

$n_gain[sf][ch]$ - для подфрейма sf и канала ch это значение представляет коэффициент усиления. Уровень представления усиления $ngain_{rl}$ получается как:

$$ngain_{rl} = n_gain[sf][ch]$$

$n_lsf[sf][ch][i]$ - для подфрейма sf и канала ch это значение представляет номер коэффициента LSF i . Разрешенный диапазон для n_lsf составляет [0,255]. Параметры деквантизованного LSF $nlsf_q$ получают так:

$$nlsfq [i] = n_lsf [sf] [ch] [i] * \pi / 256 + \pi / 512.$$

$n_delta_lsf [sf] [ch] [i]$ - для подфрейма sf и канала ch это значение представляет номер дифференциального коэффициента LSF i . Они получаются, используя следующий алгоритм

для $i > 0$:

$$nlsfq [i] = (nlsfq [i-1] + n_delta_lsf [sf] [ch] [i]) * \pi / 256 + \pi / 512.$$

$n_delta_lar_den_coarse [sf] [ch] [i]$ - для подфрейма sf и канала ch это представляет номер дифференциального коэффициента LAR знаменателя i .

$n_delta_lar_den_fine [sf] [ch] [i]$ - для подфрейма sf и канала ch это представляет более высокий уровень детализации для грубого параметра коэффициента LAR знаменателя. Уровень представления $ndlar_{rl}$ является суммой дифференциального грубого LAR знаменателя и дифференциального точного LAR знаменателя:

$$ndlar_{rl} [i] = n_delta_lar_den_coarse [sf] [ch] [i] + n_delta_lar_den_fine [sf] [ch] [i]$$

и преобразовывается в коэффициент LAR согласно:

$$nlar_{rl} [i] = nlar'_{rl, psf} [i] + ndlar_{rl} [i],$$

где $nlar_{rl} [i]$ и $nlar'_{rl, psf} [i]$ представляют уровень представления LAR текущего и предыдущего подфреймов, соответственно. В случае, когда $n_laguerre_granularity$ изменяется от %1 до %0 проходя от подфрейма $sf-1$ к sf , значение $nlar'_{rl, psf} [i]$ сначала преобразовывается в наиболее грубую сетку квантования согласно:

$$rl, psf [i] = 4 * \lfloor nlar'_{rl, psf} [i] / 4 + 0,5 \rfloor,$$

где $nlar'_{rl, psf} [i]$ представляет уровень представления LAR предыдущего подфрейма.

$n_delta_gain [sf] [ch]$ - для подфрейма sf и канала ch это значение представляет дифференцированный коэффициент усиления и преобразовывается в уровень представления $ngain_{rl}$ согласно:

$$ngain_{rl} = ngain_{rl, p4sf} + n_delta_gain [sf] [ch],$$

где $ngain_{rl, p4sf}$ представляет уровень представления усиления для подфрейма $sf-4$.

$n_overlap_lsf$ - Один бит, указывающий, перекрываются ли коэффициенты LSF из предыдущего определения в канале ch .

$enable_ps_header$ - один бит, указывающий, присутствует ли информация заголовка PS . Если установлен в %1, данные заголовка PS , конфигурирующие декодер PS , передаются. Иначе, сохраняется последняя конфигурация.

$enable_iid$ - один бит, обозначающий присутствие параметров IID . Если $enable_iid$ устанавливается в %1, параметры межканального различия интенсивности (IID) будут

отправлены с этого момента в потоке битов. Если *enable_iid*==%0, никакие параметры *IID* не будут отправлены с этого момента в потоке битов.

iid_mode - конфигурация параметров *IID* (число полос и сетка квантования, *iid_quant*) определяется *iid_mode*. Поддерживаются восемь различных конфигураций для параметров *IID* (таблица 24).

Таблица 24 - Конфигурации режима *IID*

<i>iid_mode</i>	<i>nr_iid_par_tab</i>	<i>nr_ipdopd_par_tab</i>	<i>iid_quant</i>	Диапазон индекса
0 (000)	10	5	0	-7 ... 7
1 (001)	20	11		-7 ... 7
2 (010)	34	17		-7 ... 7
3 (011)	10	5	1	-15 ... 15
4 (100)	20	11		-15 ... 15
5 (101)	34	17		-15 ... 15
6 (110)	Зарезервировано			
7 (111)	Зарезервировано			

Если данные *IID* не отправляются в потоке битов, все параметры *IID* сбрасываются в 0 (то есть *index*=0).

Сетки значения по умолчанию и тонкого квантования для *IID*, *iid_quant* = %0 и *iid_quant* = %1 назначаются, как предусмотрено в таблице 25 и таблице 26, соответственно.

Таблица 25 - Сетка квантования по умолчанию для *IID*.

Индекс	-7	-6	-5	-4	-3	-2	-1	0
<i>IID</i> [дБ]	-25	-18	-14	-10	-7	-4	-2	0
Индекс	1	2	3	4	5	6	7	
<i>IID</i> [дБ]	2	4	7	10	14	18	25	

Таблица 26 - Сетка тонкого квантования для *IID*.

Индекс	-15	-14	-13	-12	-11	-10	-9	-8
<i>IID</i> [дБ]	-50	-45	-40	-35	-30	-25	-22	-19
Индекс	-7	-6	-5	-4	-3	-2	-1	0
<i>IID</i> [дБ]	-16	-13	-10	-8	-6	-4	-2	0
Индекс	1	2	3	4	5	6	7	8
<i>IID</i> [дБ]	2	4	6	8	10	13	16	19
Индекс	9	10	11	12	13	14	15	
<i>IID</i> [дБ]	22	25	30	35	40	45	50	

Конфигурация параметров межканальной разности фаз (*IPD*) / полной разности фаз (*OPD*), строго связывается с конфигурацией *IID*. Это также иллюстрируется в таблице 24.

enable_icc - один бит, обозначающий присутствие параметров *ICC*. Если *enable_icc* будет установлен в %1, то параметры межканальной когерентности (*ICC*) будут отправлены с этого момента в потоке битов. Если *enable_icc* = %0, никакие параметры *ICC* не будут отправлены с этого момента в потоке битов.

icc_mode - конфигурация параметров межканальной когерентности (число полос и сетка квантования) определяется *icc_mode*. Для параметров *ICC* поддерживаются восемь различных конфигураций (таблица 27).

Таблица 27 - Конфигурация режима *ICC*

<i>icc_mode</i>	<i>nr_icc_par_tab</i>	Диапазон индекса	Процедуры смешивания
0 (000)	10	0 ... 7	R_a
1 (001)	20	0 ... 7	
2 (010)	34	0 ... 7	
3 (011)	10	0 ... 7	R_b
4 (100)	20	0 ... 7	
5 (101)	34	0 ... 7	
6 (110)	зарезервировано		
7 (111)	зарезервировано		

Если никакие данные *ICC* не отправляются в потоке битов, все параметры *ICC* сбрасываются в 1 (то есть *index*=0). Сетка квантования по умолчанию для *ICC* приводится в таблице 28.

Таблица 28 - Сетка квантования для ICC.

Индекс	0	1	2	3	4	5	6	7
ρ	1	0,937	0,84118	0,60092	0,36764	0	-0,589	-1

enable_ext - уровень расширения PS задействуется, используя бит *enable_ext*. Если он устанавливается в %1, параметры *IPD* и *OPD* отправляются. Если это отключается, то есть %0, уровень расширения пропускается, и *enable_ipdopd* устанавливается в %0.

frame_class - бит *frame_class* определяет, располагаются ли позиции параметров текущего фрейма равномерно на протяжении фрейм (*FIX_BORDERS: frame_class == %0*) или они определяются, используя позиции, описанные *border_position* (*VAR_BORDERS: frame_class == %1*).

num_env_idx - число (наборов) параметров (огигающих) на фрейм определяется, используя *num_env_idx*. В случае фиксированного интервала параметров (*frame_class == %0*) и переменного интервала параметров (*frame_class == %1*) это соотношение показано в таблице 29.

num_env - локальная переменная, обозначающая число огигающих стерео (наборов параметров). *num_env == 0* сигнализирует о том, что никакие новые параметры стерео не передаются, и что последние параметры в предыдущем элементе *ps_data ()* должны сохраняться неизменными и применяться к текущему элементу *ps_data ()*.

Таблица 29 - Число наборов параметров *num_env* как функция *num_env_idx* в случае фиксированного и переменного интервала.

<i>num_env_idx</i>	<i>num_env_tab [frame_class] [num_env_idx]</i>	
	<i>frame_class == 0</i>	<i>frame_class == 1</i>
0	0	1
1	1	2
2	2	3
3	4	4

border_position [e] - в случае расположения параметров с переменными интервалами позиции параметров определяются *border_position [e]*. Он содержит индекс выборки *QMF n_e* для набора параметров *e* текущего элемента *ps_data ()*.

iid_dt [e] - этот флаг описывает для индекса огигающей *e*, кодируются ли параметры IID дифференцированно в течение времени (*iid_dt == %1*) или по частоте (*iid_dt == %0*). В случае, когда *iid_mode* текущего элемента *ps_data ()* отличается от *iid_mode*

предыдущего элемента *ps_data* (), *iid_dt* [0] должно иметь значение 0%, принуждая к дифференцированному кодированию частоты.

iid_data () - синтаксический элемент, содержащий данные *IID*.

icc_dt [*e*] - этот флаг описывает для индекса огибающей *e*, кодируются ли параметры *ICC* дифференцированно в течение времени (*icc_dt* == %1) или по частоте (*icc_dt* == %0). В случае, когда *icc_mode* текущего элемента *ps_data* () отличается от *icc_mode* предыдущего элемента, *ps_data* (), *icc_dt* [0] должно иметь значение 0%, принуждая к дифференцированному кодированию частоты.

icc_data () - синтаксический элемент, содержащий данные *ICC*.

cnt - локальная переменная, обозначающая число байтов, используемых для элемента *ps_extension* ().

ps_extension_size - длиной уровня расширения *PS* является *ps_extension_size*, измеренный в байтах. Если размер расширения приводит к использованию кода *escape* (*ps_extension_size* == 15), длина уровня расширения увеличивается на дополнительное количество байтов.

esc_count - в случае, если используется код *escape* (*ps_extension_size* == 15), *esc_count* описывает дополнительную длину уровня расширения *PS*, измеренную в байтах.

num_bits_left - локальная переменная, описывающая число битов, остающихся для чтения в элементе *ps_extension* ().

ps_extension_id - идентификационная бирка (версия) уровня расширения *PS* дается *ps_extension_id*. В данный момент поддерживается только одна версия (таблица 30).

Таблица 30 - Описание *ps_extension_id*

<i>ps_extension_id</i>	Версия
00 (0)	v0
01 (1)	Зарезервировано
10 (2)	Зарезервировано
11 (3)	Зарезервировано

fill_bits - Эти *fill_bits* выполняют побайтное выравнивание данных *ps_extension* ().

enable_ipdopd - применение параметров *IPD* и *OPD* в потоке битов обозначается *enable_ipdopd*. Если установлено (*enable_ipdopd* == %1), параметры *IPD* и *OPD* отправляются, если отключено (*enable_ipdopd* == %0), параметры *IPD* и *OPD* для текущего фрейма в потоке битов не отправляются. В случае *enable_iid* == %0, *enable_ipdopd* не должен быть установлен в %1. Сетка квантования и для *IPD*, и для *OPD* обеспечивается

в таблице 31. Если данные *IPD* или *OPD* не отправляются в потоке битов, все параметры *IPD* и *OPD* устанавливаются в 0 (то есть *index=0*).

Таблица 31 - Сетка квантования для *IPD/OPD*

Индекс	0	1	2	3	4	5	6	7
Уровень представления	0	$\frac{\rho}{4}$	$\frac{\rho}{2}$	$\frac{3\rho}{4}$	π	$\frac{5}{4}\rho$	$\frac{3}{2}\rho$	$\frac{7}{4}\rho$

ipd_dt [e] - этот флаг описывает для индекса огибающей *e*, кодируются ли параметры *IPD* дифференцированно в течение времени (*ipd_dt == %1*) или по частоте (*ipd_dt == %0*). В случае, когда *iid_mode* текущего элемента *ps_data ()* отличается от *iid_mode* предыдущего элемента *ps_data ()*, у *ipd_dt [0]* должно быть значение 0%, принуждающее к измененному кодированию частоты.

ipd_data () - синтаксический элемент, содержащий данные *IPD*.

opd_dt [e] - этот флаг описывает для индекса огибающей *e*, кодируются ли параметры *OPD* дифференцированно в течение времени (*opd_dt == %1*) или по частоте (*opd_dt == %0*). В случае, когда *iid_mode* текущего элемента *ps_data ()* отличается от *iid_mode* предыдущего элемента *ps_data ()*, у *opd_dt [0]* должно быть значение 0%, принуждающее к иному кодированию частоты.

opd_data () - синтаксический элемент, содержащий данные *OPD*.

reserved_ps - этот бит зарезервирован и имеет значение % 0.

iid_par_dt [e] [b] - в случае дифференцированного кодирования параметров *IID* в течение времени (*iid_dt [e] == %1*), *iid_par_dt [e] [b]* описывает различие индекса *IID* относительно позиции параметра b^{th} для огибающей *e-1*. Если никакой предыдущий параметр не доступен, *iid_par_dt [e] [b]* представляет различие индекса *IID* относительно декодируемого значения 0 (т.е. *index=0*). Индекс *IID* *iid_par [e] [b]*, определяется как:

$$iid_par [e] [b] = iid_par [e - 1] [b] + iid_par_dt [e] [b],$$

где *iid_par [e-1] [b]* представляет индекс *IID* предыдущей огибающей *e-1*. Значение *IID*, *iid [b]*, получается при использовании *iid_par [e] [b]*, как индекса к таблице 25 или таблице 26, в зависимости от *iid_mode*.

iid_par_df [e] [b] - в случае дифференцированного кодирования параметров *IID* по частоте (*iid_dt [e] == %0*), *iid_par_df [e] [b]* описывает отличие *IID* относительно (*b-1*) - го параметра в огибающей *e*. Если никакие предыдущие параметры не доступны, *iid_par_df [e] [b]* представляет отличие *IID* относительно декодируемого значения 0 (то есть *index=0*). Индекс *IID*, *iid_par [e] [b]*, определяется как:

$$iid_par [e] [0] = iid_par_df [e] [0],$$

$$iid_par [e] [b] = iid_par [e] [b-1] + iid_par_df [e] [b] \quad \text{для } b > 0,$$

где $iid_par [e] [b-1]$ представляет индекс IID предыдущего значения IID для огибающей e . Величина IID , $iid [b]$, получается при использовании $iid_par [e] [b]$, как индекса к таблице 25 или таблице 26, в зависимости от iid_mode .

$icc_par_dt [e] [b]$ - в случае дифференцированного кодирования параметров ICC в течение времени ($icc_dt [e] == \%1$), $icc_par_dt [e] [b]$ описывает отличие относительно b^{or} позиции параметра для огибающей $e-1$. Если никакой предыдущий параметр не доступен, $icc_par_dt [e] [b]$ представляет различие в ICC относительно декодируемого значения 1 (то есть $index=0$). Индекс ICC , $icc_par [e] [b]$ определяется как:

$$icc_par [e] [b] = icc_par [e-1] [b] + icc_par_dt [e] [b],$$

где $icc_par [e-1] [b]$ представляет индекс ICC предыдущей огибающей $e-1$. Значение ICC , $\rho [b]$ получается при использовании $icc_par [e] [b]$, как индекса к таблице 28.

$icc_par_df [e] [b]$ - в случае дифференцированного кодирования параметров ICC по частоте ($icc_dt [e] == \%0$), $icc_par_df [e] [b]$ описывает отличие ICC относительно $(b-1)^{to}$ параметра для огибающей e . Если предыдущие параметры не доступны, $icc_par_df [e] [b]$ представляет отличие ICC относительно декодируемого значения 1 (то есть $index=0$). Индекс ICC , $icc_par [e] [b]$ определяется как:

$$icc_par [e] [0] = icc_par_df [e] [0],$$

$$icc_par [e] [b] = icc_par [e] [b-1] + icc_par_df [e] [b] \quad \text{для } b > 0,$$

где $icc_par [e] [b-1]$ представляет индекс ICC предыдущего значения ICC для огибающей e . Значение ICC $\rho [b]$ получается при использовании $icc_par [e] [b]$, как индекса к таблице 28.

$ipd_par_dt [e] [b]$ - в случае дифференцированного кодирования параметров IPD в течение времени ($ipd_dt [e] == \%1$), $ipd_par_dt [e] [b]$ описывает отличие IPD относительно b^{or} позиции параметра для огибающей e . Если никакой предыдущий параметр не доступен, $ipd_par_dt [e] [b]$ представляет отличие IPD относительно декодируемого значения 0 (то есть $index=0$). Для параметров IPD по модулю 8 применяется дифференцированное кодирование. Индекс IPD , $ipd_par [e] [b]$, определяется как:

$$ipd_par [e] [b] = mod(ipd_par [e-1] [b] + ipd_par_dt [e] [b], 8)$$

где $ipd_par [e-1] [b]$ представляет индекс IPD предыдущей огибающей, $e-1$. Величина IPD , $ipd [b]$, получается при использовании $ipd_par [e] [b]$, как индекса к таблице 31.

$ipd_par_df [e] [b]$ - в случае дифференцированного кодирования параметров IPD по частоте ($ipd_dt [e] == \%0$), $ipd_par_df [e] [b]$ описывает различие IPD относительно

$(b-1)^{то}$ параметра для огибающей e . Если предыдущие параметры не доступны, $ipd_par_df [e] [b]$ представляет отличие IPD относительно декодируемого значения 0 (то есть $index=0$). Для параметров IPD по модулю 8 применяется дифференцированное кодирование. Индекс IPD , $ipd_par [e] [b]$, определяется как:

$$ipd_par [e] [0] = ipd_par_df [e] [0] [e] [0] ,$$

$$ipd_par [e] [b] = mod(ipd_par[e][b-1] + ipd_par_df [e] [b] , 8) \text{ для } b > 0,$$

где $ipd_par [e] [b-1]$ представляет индекс IPD предыдущего значения IPD для огибающей e . Значение IPD , $ipd [b]$ получается при использовании $ipd_par [e] [b]$, как индекса к таблице 31.

$opd_par_dt [e] [b]$ - в случае дифференцированного кодирования параметров OPD в течение времени ($opd_dt [e] == \%1$), $opd_par_dt [e] [b]$ описывает отличие OPD относительно $b^{ой}$ позиции параметра для огибающей ($e-1$). Если предыдущие параметры не доступны, $opd_par_dt [e] [b]$ представляет отличие OPD относительно декодируемого значения 0 (то есть $index=0$). Замечание: для параметров OPD по модулю 8 применяется дифференцированное кодирование. Индекс OPD , $opd_par [e] [b]$, определяется как:

$$opd_par [e] [b] = mod(opd_par[e-1][b] + opd_par_dt[e][b], 8) ,$$

где $opd_par [e-1] [b]$ представляет индекс OPD предыдущей огибающей, $e-1$. Значение OPD , $opd [b]$, получается при использовании $opd_par [e] [b]$, как индекса к таблице 31.

$opd_par_df [e] [b]$ - в случае дифференцированного кодирования параметров OPD в течение времени ($opd_dt [e] == \%0$), $opd_par_df [e] [b]$ описывает отличие OPD относительно $(b-1)^{ой}$ позиции параметра для огибающей e . Если предыдущие параметры не доступны, $opd_par_df [e] [b]$ представляет отличие OPD относительно декодируемого значения 0 (то есть $index=0$). Для параметров OPD по модулю 8 применяется дифференцированное кодирование. Индекс OPD , $opd_par [e] [b]$, определяется как:

$$opd_par [e] [0] = opd_par_df [e] [0]$$

$$opd_par [e] [b] = mod (opd_par [e] [b-1] + opd_par_df [e] [b] , 8) \text{ для } b > 0$$

где $opd_par [e] [b-1]$ представляет индекс OPD предыдущего значения OPD для огибающей e . Величина OPD , $opd [b]$, получается при использовании $opd_par [e] [b]$, как индекса к таблице 31.

5.3 Индексация подфреймов

В случае, когда дифференцированное кодирование применяется от одного подфрейма до следующего, может быть назначен отрицательный индекс подфрейма sf фрейма k . В этом случае отрицательный подфрейм должен быть скорректирован со-

гласно $sf = sf + nrof_subframes$.

Полученный таким образом индекс подфрейма находится во фрейме $k-1$. Точно так же в случае, когда sf больше, чем $nrof_subframes$, подфрейм должен быть скорректирован согласно $sf = sf - nrof_subframes$.

Индекс подфрейма, полученный таким образом, находится во фрейме $k+1$.

6 Процесс декодирования

После деформатирования потока битов монофонический сигнал M восстанавливается как комбинация переходных процессов, синусоид и шума. Впоследствии используются параметры стерео, чтобы восстановить левый и правый сигналы из монофонического кодированного сигнала. Для двойного моно и моно параметрический декодер стерео не используется.

Параметрический декодер состоит из трех декодеров: декодера переходов, синусоидального декодера и декодера шума. Декодируемый сигнал получается суммированием выходов этих трех декодеров. В описании декодеров в параметрическом декодере индексация подфрейма sf и канала ch иногда для ясности опускается.

6.1 Переходные процессы

Определяются два типа переходных процессов, переходный процесс шага и переходный процесс типа Мейклера. Декодирование переходного процесса шага включает только интерпретацию позиции. Для типа Мейклера должны декодироваться параметризованная огибающая $g[n]$ и ряд синусоид.

6.1.1 Переходный процесс шага

Переходный процесс шага не генерирует собственный сигнал, но он используется, чтобы изменить форму окна для того, чтобы синтезировать синусоидальные и шумовые компоненты.

6.1.2 Переходный процесс Мейклера

Для декодирования переходного процесса Мейклера сначала должна быть сгенерирована огибающая. Для огибающей требуются следующие параметры: стартовая позиция t_{loc} , начальный наклон, обозначенный t_b_par (параметр b) и наклон спада, представленный t_chi_par (параметр ξ) (рисунок 1).

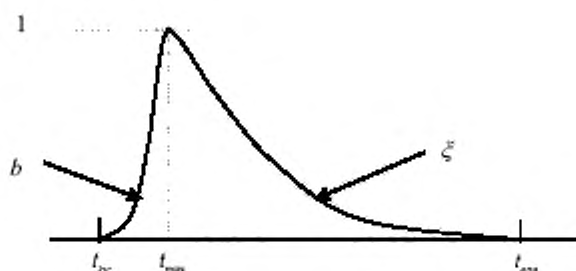


Рисунок 1 – Огибающая Мейклера описывается функцией $g[n]$. Наклоном атаки управляет параметр b . Спадом управляет параметр ξ

Время старта огибающей t_{loc} для простоты объяснения принимается как $n = 0$. Эта огибающая $g[n]$ сгенерирована согласно

$$g[0] = \frac{(1 - x^2)^{\frac{b}{2}}}{a_{max}},$$

$$g[n] = g[n-1] \chi \sqrt{\frac{b+n-1}{n}},$$

для $n=1$, включая t_{end} , конечная позиция переходного окна t_{end} определяется ниже. Максимум a_{max} находится в позиции t_{max} и дается приближениями

$$t_{max} = \frac{b-1}{-2 \chi \log_e(x)},$$

$$a_{max} = \sqrt{\frac{-2 \chi \log_e(x)}{\Gamma(b)}} \chi \left(\frac{b-1}{e}\right)^{\frac{b-1}{2}}.$$

Эти сложные выражения, особенно выражение для a_{max} , были оценены для допустимых значений t_{b_par} и t_{chi_par} и сводятся в таблицу 32.

Таблица 32 - t_{max} для всех возможных значений t_{b_par} и t_{chi_par}

t_{max}		t_{b_par}			
		0	1	2	3
t_{chi_par}	0	15	30	45	59
	1	20	39	57	75
	2	30	56	79	100
	3	57	96	126	150

Таблица 33 - a_{max} для всех возможных значений t_b_par и t_chi_par

a_{max}		t_b_par			
		0	1	2	3
t_chi_par	0	0.152713500109658	0.131630525645664	0.120142673294398	0.112550174511598
	1	0.132843681407528	0.115639700421076	0.106510539071702	0.100663024431527
	2	0.109279971016712	0.0971964875412947	0.0909719057150294	0.0872632874594248
	3	0.0797175749717262	0.0744985442180281	0.0723059623257423	0.0715041477354716

Переходный процесс должен быть синтезирован до t_{end} . Позиция t_{end} определяется как конец второго полного последующего подфрейма после позиции переходного процесса t_{loc} .

$$t_{end} = 3\text{С} - t_{loc} - 1$$

Подсчет выборок в подфрейме начинается с 0. Если переходный процесс стартует точно в начале подфрейма ($t_{loc} = 0$), то $t_{end} = 3\text{С} - 1$.

6.1.3 Синусоиды под огибающей

Получающееся представление переходного процесса получается путем комбинирования огибающей и синусоид согласно

$$t[n] = g[n] \prod_{i=1}^{t_cont_sin} ta_q[i] \cos(tf_q[i]) + tp_q[i],$$

для n вплоть до $n=0$ и включая t_{end} .

6.2 Синусоиды

6.2.1 Соединение

Фрейм обновления, обозначенный `refresh_sinusoids == %1`, используется, чтобы указать, стартует ли фрейм с абсолютных значений для всех продолжений или стартует с дифференциально кодированных продолжений. Для рождения каждой синусоидальной дорожки во фрейме в потоке битов предоставляется `s_cont`, чтобы сигнализировать о числе подфреймов, в которых продолжается дорожка после текущего подфрейма в этом фрейме. В случае, если дорожка продолжается в первом подфрейме следующего фрейма, к этому числу добавляется 1. Если дорожка продолжается после первого подфрейма следующего фрейма, к этому числу добавляется 2. Основываясь на этой информации, декодер в состоянии соединить параметры, которые принадлежат дорожке.

`s_cont` заполняется в следующем порядке:

- 1) продолжения;
- 2) рождения (отсортированные по частоте в порядке возрастания).

При прохождении от одного подфрейма к следующему декодер отслеживает число продолжений, $s_nrof_continuations [sf]$. Число продолжений, существующих в подфрейме $sf+1$, может быть непосредственно получено из числа записей в $s_cont [sf]$, не равных нулю. Для первого подфрейма в фрейме $s_nrof_continuations$ читается из потока битов, чтобы задействовать произвольный доступ.

Общая количество синусоидальных компонентов в подфрейме sf , $s_nrof_sin [sf]$, вычисляется так:

$sf == 0$:

$s_nrof_sin[0] = s_nrof_continuations[0][ch] + s_nrof_births[0][ch]$,

$sf > 0$

$s_nrof_sin[sf] = \prod_{i=1}^{\max_nrof_sin[sf-1]} (s_cont[sf-1][ch][i] > 0) + s_nrof_births[sf][ch]$

6.2.2 Декодирование синусоидальных параметров

В описании ниже мы принимаем, что имеет место синусоидальная дорожка длины k , в подфреймах $sf = [K, K_{sk} + 1]$. Для рождений дорожки ($sf = K$) частота и фаза индекса синусоиды n представляются $f_q [K] [ch] [n]$ и $sp_q [K] [ch] [n]$, соответственно. Для продолжений, чтобы получить информацию о частоте и фазе для подфрейма, требуются уровни представления вместе с отслеживанием информации.

Для продолжения ($sf = [K+1, K_{sk} - 1]$) уровни представления $s_delta_cont_freq_pha [sf] [ch] [n]$ преобразовываются в квантованную ошибку прогноза $\Delta [sf] [ch] [n]$, используя таблицу 35 с индексом $= 2$. Квантованная ошибка прогноза $\Delta [sf] [ch] [n]$ затем умножается на масштабный коэффициент $c [sf] [ch] [n]$. Для первого продолжения $c [K+1] [ch] [p]$, где p представляет индекс синусоиды в подфрейме $K+1$, зависит от частоты рождения, то есть $f_q [K]$. Таблица 34 показывает значение масштабного коэффициента c для возможных частотных диапазонов $f_q [K]$.

Таблица 34 - Таблица масштабных коэффициентов

Частотный диапазон, Гц	Масштабный коэффициент $c[K+1]$
[0 - 500]	1/8
<500-1000]	1/4
<1000-4000]	1/2
<4000-22050]	1

Для других продолжений $sf = [K+2, K_{sk}-1]$ с изменяется согласно полученным уровням представления вдоль дорожки посредством блока “Backward Adaptation”. Если $s_delta_cont_freq_pha [sf] [ch] [n]$ равно 1 или 2 (внутренний уровень) для подфрейма sf , то c для подфрейма $sf+1$ устанавливается в

$$c[sf+1][ch][p] = \{s\}[ch][n] \cdot \tilde{x}^{1/4}.$$

Если $s_delta_cont_freq_pha [sf] [ch] [n]$ равняется 0 или 3 (внешний уровень), то c для подфрейма $sf+1$ устанавливается в

$$c[sf+1][ch][p] = \{s\}[ch][n] \cdot \tilde{x}^{1/2}.$$

Чтобы избежать очень маленьких или очень больших записей по ошибке прогноза, адаптация выполняется только если абсолютное значение внутреннего уровня $0,75c [sf+1] [ch] [p]$ находится между $\pi/128$ и $3\pi/8$.

При получении квантованной ошибки прогноза Δ вывод фильтра прогноза добавляется к ней, приводя к развернутой фазе ψ_r :

$$\psi_r [sf] [ch] [n] = \psi_p [sf] [ch] [n] + \Delta [sf] [ch] [n] \cdot c [sf] [ch] [n].$$

Используется прогнозирующее устройство второго порядка. Поведение ввода - вывода фильтра таково

$$\psi_p [sf+1] [ch] [p] = 2 \cdot \psi_r [sf] [ch] [n] - \psi_r [sf-1] [ch] [q],$$

где q является индексом синусоиды в подфрейме $sf-1$, ψ_r является вводом и ψ_p является выводом фильтра прогноза.

Чтобы инициализировать фильтр прогноза, нужно одно значение из предыдущих значений для ввода $\psi_r [K-1]$, где K является индексом подфрейма рождения дорожки. Так как это значение не доступно, предполагаем, что частота является постоянной в подфрейме $(K-1)$. Для рождения дорожки имеем информацию о частоте и фазе, таким образом, можем вычислить ввод в $(K-1)$ и K согласно:

$$\psi_r [K-1] [ch] [n] = sp_q [K] [ch] [n] - f_q [K] [ch] [n] S,$$

$$\psi_r [K] [ch] [n] = sp_q [K] [ch] [n],$$

где S представляет интервал обновления. Развернутые фазы являются отфильтрованными по низким частотам *LPF-phase* блока. Это делается следующим образом:

$$\psi [sf] [ch] [n] = 0,25\psi_r [sf+1] [ch] [p] + 0,5\psi_r [sf] [ch] [n] + 0,25\psi_r [sf-1] [ch] [q],$$

где sf является индексом подфрейма вдоль дорожки. В конце дорожки ($sf = K_{sk}-1$), применяется следующее правило:

$$\psi [sf] [ch] [n] = \psi_r [sf] [ch] [n].$$

Восстановленные фазы получают из сглаженных развернутых фаз следующим

образом:

$$sp_q [sf] [ch] [n] = \text{mod} (\psi [sf] [ch] [n] + \pi, 2) - \pi.$$

Чтобы получить частоту, развернутые фазы должны дифференцироваться вдоль дорожки. Дифференциация реализуется приближением. Частота получается с помощью:

$$v[sf][ch][n] \approx \frac{\partial \psi[sf][ch][n]}{\partial sf} \cdot \frac{2}{S} - v[sf-1][ch][q],$$

где S представляет интервал обновления и $\varpi [K] [ch] [n] = f_q [K] [ch] [n]$. Когда фаза рождения и частота рождения известны в декодере, вычисляются частоты ϖ последующих фреймов. Чтобы ослабить сигнал с шумами, которые вносятся этой дифференциацией, применяется фильтр низких частот на частоты (*LPF-freq*):

$$f_q [sf] [ch] [n] = 0,25 \varpi [sf+1] [ch] [p] + 0,5 \varpi [sf] [ch] [n] + 0,25 \varpi [sf-1] [ch] [q].$$

Для первого продолжения дорожки ($sf=K+1$) определение изменяется на:

$$f_q [K+1] [ch] [p] = 0,5 \cdot \varpi [K+1] [ch] [p] + 0,5 \cdot \varpi [K] [ch] [n].$$

Также последняя частота в дорожке ($sf = K_{*k}-1$) получается другим способом:

$$f_q [sf] [ch] [n] = 0,5 \cdot \varpi [sf] [ch] [n] + 0,5 \cdot \varpi [sf-1] [ch] [q].$$

Для дорожек длиной $k=2$ продолжение вычисляется согласно

$$f_q [sf] [ch] [n] = 0,5 \cdot \varpi [sf-] [1 ch] [q] + 0,5 \cdot \varpi [sf] [ch] [n].$$

Таким образом получаются фазы и частоты из уровней представления *s_delta_cont_freq pha*.

Во фреймах обновлений применяется следующая процедура. Если $sf = [K, \dots, K+R, \dots, K_{*k}-1]$, подфрейм $K+R$ является первым подфреймом фрейма с *refresh_sinusoids == %1*. Подфрейм $(K+R-1)$ является последним подфреймом фрейма с *refresh_sinusoids_next_frame == %1*. Значения фазы и частоты для подфрейма с подфрейма K до и включая подфрейм $K+R-1$ получаются как описано выше, как будто дорожка заканчивается в подфрейме $(K+R-1)$. Значения фазы и частоты для подфрейма $K+R$ до подфрейма $(K_{*k}-1)$ получаются как описано выше, как будто подфрейм $K+R$ является рожденным. Для инициализации квантованной ошибки прогноза Δs_adpcm_grid используется в качестве индекса к таблице 35 и $c [K+R] [ch] [n] = 1$.

Таблица 35 - Квантованная ошибка прогноза Δ

Δ	<i>s_delta_cont_freq pha</i>			
	0	1	2	3
0	-4,2426	-1,0607	1,0607	4,2426
1	-3,5676	-0,8919	0,8919	3,5676
2	-3,0000	-0,7500	0,7500	3,0000
3	-2,5227	-0,6307	0,6307	2,5227
4	-2,1213	-0,5303	0,5303	2,1213
5	-1,7838	-0,4460	0,4460	1,7838
6	-1,5000	-0,3750	0,3750	1,5000
7	-1,2613	-0,3153	0,3153	1,2613
8	-1,0607	-0,2652	0,2652	1,0607
9	-0,8919	-0,2230	0,2230	0,8919
10	-0,7500	-0,1875	0,1875	0,7500
11	-0,6307	-0,1577	0,1577	0,6307
12	-0,5303	-0,1326	0,1326	0,5303
13	-0,4460	-0,1115	0,1115	0,4460
14	-0,3750	-0,0938	0,0938	0,3750
15	-0,3153	-0,0788	0,0788	0,3153
16	-0,2652	-0,0663	0,0663	0,2652
17	-0,2230	-0,0557	0,0557	0,2230
18	-0,1875	-0,0469	0,0469	0,1875
19	-0,1577	-0,0394	0,0394	0,1577
20	-0,1326	-0,0331	0,0331	0,1326
21	-0,1115	-0,0279	0,0279	0,1115

В случае, когда *phase_jitter_present* устанавливается в %1, и фаза и частота получаются посредством декодирования *ADPCM*, никакое дрожание фазы не применяется.

6.2.3 Синтез синусоид для сегментов без переходного процесса

Параметры синусоид используются, чтобы синтезировать синусоидальные компоненты. Это делается на основе сегмента, состоящего из выборки *L*.

Синтез использует стратегию 50% перекрытия и добавления. Чтобы синтезировать подфрейм, в начале нового фрейма должны быть доступными параметры предыдущего подфрейма. Это означает, что должны быть доступными параметры последнего

подфрейма в предыдущем фрейме.

Фактический синтез синусоиды выполняется согласно

$$s_i[n] = s_{q_i}[i] \cos(f_q[i] \psi n - \frac{L-1}{2}) + s_{p_i}[i], \text{ для } n = [0, L-1]$$

Фазовая информация s_{p_i} определяется для середины сегмента $(=(L-1)/2)$. Фазовая информация исходной фазы s_{p_i} вычисляется из s_phi (см. 5.2). Для продолжений фаза вычисляется как описано в 6.2.2.

В перекрытии и добавлении выбирается следующее, дополнительное симметричное окно амплитуды

$$w_s[n] = 1/2 - 1/2 \cos(\frac{p(2n+1)}{L}), \text{ для } n = [0, L-1].$$

Сегмент длины L получается из

$$s_{sf}[n] = w_s[n] \sum_{i=0}^{s_nrof_sin[sf]} s_i[n], \text{ для } n = [0, L-1].$$

Синусоидальное добавление для подфрейма sf тогда вычисляется, используя добавление из предыдущего подфрейма согласно :

$$s_{sf}[n] = s_{sf-1}[S+n] + s_{sf}[n], \text{ для } n = [0, S-1].$$

6.2.4 Синтез синусоид для сегментов без переходного процесса

Для синусоид с частотой ниже 400 Гц должно использоваться только окно $w_S[n]$.

Для других синусоид для синтеза сегмента применяются другие формы окна, в зависимости от компонента (продолжение, рождение, или отмирание).

Следующее выражение точно описывает окно $w_S[n]$

$$w_S[n] = \begin{cases} 0 & , 0 \leq n < d-10 \\ \frac{n-d+11}{22} & , d-10 \leq n \leq d+10 \\ 1 & , d+10 < n < S \\ 1/2 - 1/2 \cos(p \psi \frac{2n+1}{L}) & , S \leq n < L \end{cases}$$

6.2.4.2 Отмирание и переходный процесс во второй половине сегмента

Следующее выражение описывает окно $w_S[n]$

$$w_s[n] = \begin{cases} 1/2 - 1/2 \cos\left(\rho \frac{2n+1}{L}\right) & , 0 \leq n < S \\ 1 & , S \leq n < d-10 \\ \frac{d-n+11}{22} & , d-10 \leq n \leq d+10 \\ 0 & , d+10 < n < L \end{cases}$$

6.3 Шум

Шум синтезируется в интервалах 4 подфреймов или $2L$ выборок. Модель для синтеза шума состоит из генератора псевдослучайных чисел, временного регулятора огибающей, механизма окна для перекрытия и добавления, и фильтра *IIR*.

Временная огибающая $H[n]$ представляется при использовании единственного усиления и ряда линейных спектральных частот (*LSFs*), представляющих коэффициенты *LPC*. Как усиление, так и *LSFs* обновляются один раз каждые 4 подфрейма ($2L$ выборок). Коэффициенты Лагерра представляются параметрами *LAR*. Они обновляются раз за 2 подфрейма (L выборок), то есть вдвое чаще малой частоты обновления, которая используется для временной огибающей.

Чтобы предотвратить разрывы, интервалы, которые изменяются временной огибающей, имеют перекрытие 25 %. В области перекрытия используется окно Ханнинга. Первые 4 сгенерированные подфрейма $2L$ выборок стартуют с постепенного появления, используя окно Ханнинга.

Окно w_n определяется так.

$$w_s[n] = \begin{cases} 1/2 - 1/2 \cos\left(\frac{3\rho(2n+1)}{8S}\right) & , n = \left[0, \frac{4S}{3} - 1\right] \\ 1 & , n = \left[\frac{4S}{3}, 4S - 1\right], \\ \frac{6S - n}{3} - 1 - \frac{n}{3} & , n = \left[4S, \frac{16S}{3} - 1\right]. \end{cases}$$

6.3.1 Генерация шума

Шум генерируется с помощью генератора псевдослучайных чисел, определенный линейной конгруэнтной последовательностью U

$$U[n+1] = \text{mod}(a * U[n] + c, m),$$

где $U[0]$ является начальным значением, a множитель, c инкремент и m модуль (c $m=2^{32}$). В начале декодирования начальное значение устанавливается в *channel_number* (0 = левый, 1 = правый), приводя к независимым источникам шума для каждого канала. Для генерации каждого следующего интервала шума начальное значение устанавлива-

ется в конечное значение предыдущего интервала. Алгоритм приводится ниже.

```
#define RAND_SCALE          (1/4294967296.0)
# define RAND_FACTOR      1664525L
# define RAND_OFFSET      1013904223L
Double noiseUDN (unsigned long *lp_seed)
{
    *lp_seed = (*lp_seed * RAND_FACTOR+RAND_OFFSET) & 0xFFFFFFFF;
    return *lp_seed * RAND_SCALE;
}
```

Этот алгоритм возвращает значение $U = [0,1>$. Нормальное распределение X получается добавлением 12 последовательных выборок распределения $(U-0,5)$. Для следующей выборки X используются 12 новых последовательных выборок. Используя это нормальное распределение X , посредством фильтрации шума генерируется спектральный шум.

Чтобы избежать разрывов в генерации шума, перекрывающиеся временные интервалы берутся от генератора шума холостого хода. Это реализуется, копируя начальное число из предыдущего интервала в текущий интервал во время перекрытия.

6.3.2 Временная огибающая

Временная огибающая применяется к интервалу $2L+4S/3$ выборок, которые сгенерированы генератором случайного шума. Форма временной огибающей представляется временным доменом, эквивалентным *Line Spectral Frequencies*, которые являются представлением коэффициентов *LPC*. Параметр дополнительного усиления используется, чтобы масштабировать всю огибающую. Так как у интервалов *LSF* имеется перекрытие 25 %, существует потенциальная избыточность в параметрах *LSF* в этой области перекрытия. В случае, когда присутствует эта избыточность, кодируется только один набор *LSFs*, допустимый для двух огибающих в области перекрытия. Об этой ситуации сообщается параметром *n_overlap_lsf*. В случае *refresh_noise == %1* первый *LSF* и усиление для этого определенного интервала кодируются в абсолютных значениях посредством параметров *n_lsf* и *n_gain*. Последующие *LSFs* кодируются дифференцированно со ссылкой друг на друга.

В случае *refresh_noise == %0*, параметр усиления для этого интервала кодируется дифференцированно со ссылкой на усиление предыдущего фрейма посредством параметра *n_delta_gain*. Кодирование параметров *LSF* в той ситуации зависит от установки *n_overlap_lsf*. В случае *refresh_noise == %0* и *n_overlap_lsf == %0*, *LSF* кодируется как в ситуации где *refresh_noise == %1*. В случае *refresh_noise == %0* и *n_overlap_lsf == %1*, число коэффициентов *LSF*, *n_nrof_overlap*, вычисляется из предыдущего определения в

канале ch согласно

```

n_nrof_overlap = 0;
for (i = 0; i < n_nrof_lsf; i++) {
  if (n_lsf[sf-4][ch][i] >= 192) n_nrof_overlap++;
}

```

Коэффициенты LSF , которые перекрываются, копируются из предыдущего определения согласно

```

for (i = 0, j = n_nrof_lsf - n_nrof_overlap; i < n_nrof_overlap; i++, j++)
{
  n_lsf[sf][ch][i] = n_lsf[sf-4][ch][j] - 192;
}

```

6.3.2.1 Декодирование параметров усиления и LSF

Усиление масштабирует всю временную огибающую. Существуют две огибающих на фрейм, усиление для первой и второй временной огибающей кодируется в $sf=0$ и $sf=4$ соответственно. Коэффициент усиления G , который фактически применяется к временной огибающей, вычисляется как

$$G = \begin{cases} 0 & \text{если } ngain_{ci} = 0 \\ 10 \frac{ngain_{ci} - 21}{20} & \text{в других случаях} \end{cases}$$

Декодируемые $LSFs$, $nlsf_q$ преобразовываются в a -параметры, используя следующие уравнения. Все $LSFs$ преобразовываются к позициям на единичном круге

$$z[i] = e^{j2\pi nlsf_q(i)} \quad i = [1, n_nr_of_lsf].$$

Эти позиции разделяются на два полинома:

$$z_p[i] = z[2i] \quad i = \left[\frac{K_1 - nr_of_lsf}{K_1}, \frac{K_1 - nr_of_lsf}{K_1} - \frac{K_1 - nr_of_lsf}{2} \right]$$

$$z_q[i] = z[2i - 1] \quad i = \left[\frac{K_1 - nr_of_lsf}{K_1}, \frac{K_1 - nr_of_lsf}{K_1} - \frac{K_1 - nr_of_lsf}{2} \right]$$

Для обоих полиномов комплексно сопряженные элементы связываются

$$z_p[i] = z_p^* \left[\frac{K_1 - nr_of_lsf}{K_1} - \frac{K_1 - nr_of_lsf}{2} - i \right] \quad i = \left[\frac{K_1 - nr_of_lsf}{K_1}, \frac{K_1 - nr_of_lsf}{K_1} - \frac{K_1 - nr_of_lsf}{2} \right]$$

$$z_q[i] = z_q^* \left[\frac{K_1 - nr_of_lsf}{K_1} - \frac{K_1 - nr_of_lsf}{2} - i \right] \quad i = \left[\frac{K_1 - nr_of_lsf}{K_1}, \frac{K_1 - nr_of_lsf}{K_1} - \frac{K_1 - nr_of_lsf}{2} \right]$$

Полиномы вычисляются следующим образом

$$P_p = \prod_{i=1}^{2 \frac{K_1 - nr_of_lsf}{K_1}} (z - z_p[i]),$$

$$P_Q = \prod_{i=1}^{n_nr_of_lsf} (z - z_Q[i]).$$

В случае, если $n_nr_of_lsf$ нечетен, полиномы изменяются так

$$P_Q = P_Q(z+1)(z-1).$$

В случае, если $n_nr_of_lsf$ четный

$$P_p = P_p(z-1)$$

$$P_Q = P_Q(z-1).$$

Полином $A(z)$ дается как

$$A = \frac{P_p(z) + P_Q(z)}{2}.$$

В конечном счете огибающая $H[n]$, где n является индексом выборки, вычисляется как:

$$H[n] = \frac{G}{A \cdot e^{j\left(\frac{n\rho}{2L} + \frac{4S}{\pi}\right)}}, \quad n = \left\lfloor \frac{S}{2L} \right\rfloor + \frac{4S}{\pi} \cdot \frac{1}{2}$$

Последовательность шума умножается на временную огибающую. В случае $n_nr_of_lsf == 0$, $H[n]$ определяется как

$$H[n] = G, \quad n = \left\lfloor \frac{S}{2L} \right\rfloor + \frac{4S}{3} \cdot \frac{1}{2}$$

6.3.3 Шумовая фильтрация

Фильтр синтеза Лагерра применяется после работы по перекрытию – добавлению.

Параметры для фильтрации Лагерра обновляются каждые 2 подфрейма (выборки). Чтобы удостовериться, что в генерации первой выборки уже получается требуемая спектральная плотность, должны быть установлены начальные состояния фильтра. Это реализуется копированием конечных состояний после генерации интервала в начальные состояния для генерации следующего интервала. В случае, если $refresh_noise == \%0$, или запуска декодирования, начальные состояния фильтра будут установлены в 0. Так, используя первый набор параметров для следующих 2 подфреймов (с новыми параметрами) в качестве начальных состояний, используются конечные состояния фильтра, полученные в результате его предыдущей работы. Для обновлений параметров LAR используется дифференциальное кодирование относительно параметров LAR предыдущего интервала.

В потоке битов, для каждого подфрейма, коэффициенты для структуры фильтра Лагерра кодируются в нотации *Log Area Ratio (LAR)*. Первый шаг реконструкции параметров Лагерра является деквантизацией *LARs*. Коэффициенты *parcor ('rfc')* получаются из деквантованных *LARs*. Последний шаг является преобразованием коэффициентов *FIR* обратно в коэффициенты Лагерра.

6.3.3.1 Квантованные *LARs*

Коэффициенты *LAR* в знаменателе деквантуются, умножая значение, закодированное в потоке битов на постоянное Δ_{LAR} , которое определяется как

$$D_{LAR} = \frac{dynr}{levels - 1},$$

где $dynr=2*8$ является динамическим диапазоном коэффициентов *LAR* (от -8 до +8), и $levels=2^{bits}-2$, с $bits=9$ представляет число уровней представления.

6.3.3.2 Преобразование *LARs* в *parcors*

Следующий алгоритм описывает преобразование m коэффициентов *LAR*, $nlar_q$ в m ($m=n_nrof_den$) *parcors 'rfc'*.

```
for (i=0; i < m; i++)
{
rfc [i] = (exp (nlar_q [i])-1) / (exp (nlar_q [i]) +1)
}
```

6.3.3.3 Преобразование *parcors* в коэффициенты *FIR*

Следующий алгоритм описывает преобразование m коэффициентов *parcor*, 'rfc' в $m+1$ a -параметров 'p'.

```
for (k=0; k < m; k++)
{
d [k] = - rfc [k];
for (i=0; i < k; i++)
{
d [i] = tmp [i] + rfc [k] * tmp [k-i-1];
}
for (i=0; i <= k; i++)
{
tmp [i] = d [i];
}
}
```

```

}
p [0] = 1,0;
for (k = 0; k < m.; k ++ )
{
p [k+1] = -d [k];
}

```

6.3.3.4 Преобразование коэффициентов *FIR* в коэффициенты Лагерра

a-параметры *p* преобразуются обратно в коэффициенты Лагерра *a*, используя следующий алгоритм

$$a_m = P_m,$$

$$a_m = P_m - a_{m+1} I,$$

где $m = [n_nrof_den - 1.. 0]$ и

$$a_m = \frac{a_m}{a_0} \sqrt{1 - I^2}.$$

6.4 Параметрическое стерео

6.4.1 Параметры стерео

В представлении образа стерео используются три различных типа параметров стерео:

- 1) межканальное различие в интенсивности или *IID*, определенный относительными уровнями ограниченного полосой сигнала;
- 2) межканальное и полное различие фаз, *IPD* и *OPD*, определяющее поведение фазы ограниченного полосой сигнала;
- 3) межканальная когерентность *ICC*, определяющая (не)сходство левого и правого ограниченного полосой сигнала.

Ввод в параметрический стереодекодер состоит из монофонического параметрически сгенерированного сигнала, полученного посредством синтеза переходного процесса, синусоид и шума. Вывод состоит из левого и правого представления стерео соответственно.

6.4.2 Блок фильтров анализа *QMF*

Этот блок фильтров идентичен 64 блоку фильтров комплексного анализа *QMF*. Однако в уравнении матрицы $M(k, n)$ множитель " $(2*n+1)$ " нужно заменить на " $(2*n-$

1)". Ввод в блок фильтров является блоками из 64 выборок монофонического синтезируемого сигнала M . Для каждого блока блок фильтров выводит один слот из 64 выборок QMF .

6.4.3 Фильтрация низкой частоты

Поддиапазоны QMF далее разделяются, чтобы получить более высокую разрешающую способность по частоте, задействующую надлежащий анализ и синтез стерео для более низких частот. В зависимости от числа полос стерео были определены две гибридных конфигурации. Обзор разделений и типа фильтра, который используется, чтобы выполнить разделение, см. в таблице 36.

Таблица 36 - Обзор разделения низкой частоты для доступных конфигураций

Конфигурация, число полос стерео	Поддиапазон QMF p	Число полос Q^p	Фильтр
10, 20	0	8	Тип А
	1	2	Тип В
	2	2	
34	0	12	Тип А
	1	8	
	2	4	
	3	4	
	4	4	

$$\text{Тип А: } G_q^p = g^p[n] \exp(j \frac{2p}{Q^p} (q + \frac{1}{2})(n - 6)),$$

$$\text{Тип В: } G_q^p = g^p[n] \cos(\frac{2p}{Q^p} q(n - 6)),$$

где g^p представляет фильтры прототипа в поддиапазоне QMF p . Q^p представляет число подподдиапазонов в поддиапазоне QMF p , q индекс подподдиапазона в канале QMF p и n индекс времени. Все прототипные фильтры длиной 13 имеют задержку 6 выборок QMF . Прототипные фильтры перечисляются в таблице 37 и таблице 38 для конфигураций 10, 20 и 34 полосы стерео соответственно.

Таблица 37 - Коэффициенты прототипного фильтра для фильтров, которые разделяют нижние поддиапазоны *QMF* для конфигураций 10 и 20 полос стерео

n	$g^0[n], Q^0 = 8$	$g^{1,2}[n], Q^{1,2} = 2$
0	0,00746082949812	0
1	0,02270420949825	0,01899487526049
2	0,04546865930473	0
3	0,07266113929591	-0,07293139167538
4	0,09885108575264	0
5	0,11793710567217	0,30596630545168
6	0,125	0,5
7	0,11793710567217	0,30596630545168
8	0,09885108575264	0
9	0,07266113929591	-0,07293139167538
10	0,04546865930473	0
11	0,02270420949825	0,01899487526049
12	0,00746082949812	0

Таблица 38 - Коэффициенты прототипного фильтра для фильтров, которые разделяют нижние поддиапазоны *QMF* для конфигурация 34 полосы стерео

n	$g^0[n], Q^0 = 12$	$g^1[n], Q^1 = 8$	$g^{2,3,4}[n], Q^{2,3,4} = 4$
0	0,04081179924692	0,01565675600122	-0,05908211155639
1	0,03812810994926	0,03752716391991	-0,04871498374946
2	0,05144908135699	0,05417891378782	0
3	0,06399831151592	0,08417044116767	0,07778723915851
4	0,07428313801106	0,10307344158036	0,16486303567403
5	0,08100347892914	0,12222452249753	0,23279856662996
6	0,08333333333333	0,12500000000000	0,25000000000000
7	0,08100347892914	0,12222452249753	0,23279856662996
8	0,07428313801106	0,10307344158036	0,16486303567403
9	0,06399831151592	0,08417044116767	0,07778723915851
10	0,05144908135699	0,05417891378782	0
11	0,03812810994926	0,03752716391991	-0,04871498374946
12	0,04081179924692	0,01565675600122	-0,05908211155639

Чтобы выровнять по времени все выборки, приходящие из гибридного блока фильтров, компенсируют задержку остающихся поддиапазонов QMF, которые не фильтровались. Эта задержка составляет 6 выборок поддиапазона QMF. Это означает $G_0^k(z) = z^{-6}$ для $k = [3... 63]$ (10, 20 полос стерео) или $k = [5... 63]$ (34 полосы стерео). Чтобы компенсировать полную задержку гибридного блока фильтров анализа, первые 10 наборов (6 от задержки и 4 от фильтра QMF) гибридных поддиапазонов сбрасывается и поэтому не принимается во внимание при обработке.

Результатом этой работы является слот гибридных выборок поддиапазона, состоящих из LF (низкая частота) части поддиапазона *sub* QMF и HF (высокая частота) части поддиапазона QMF.

6.4.4 Кадрирование

Один фрейм параметрического аудио включает два фрейма данных стерео. Параметры стерео в пределах фрейма стерео могут быть присвоены одному или более слотам. Границы фрейма стерео и позиции n_e слотов, которым были присвоены параметры стерео, определяют так называемые области. Параметры стерео определяются для последнего слота.

6.4.5 Декорреляция

Посредством всечастотной фильтрации и задержки, выборки поддиапазона *sub* $s_k(n)$ преобразовываются в декоррелированные выборки поддиапазона *sub* $d_k(n)$, где k представляет частоту в гибридном спектре, а n - индекс времени.

6.4.5.1 Константы

<i>DECAY_SLOPE</i> =0,05	Наклон затухания всечастотного фильтра.
<i>NR_ALLPASS_LINKS</i> =3	Число связей фильтра для всечастотного фильтра
<i>NR_PAR_BANDS</i>	Число полос частот, которые могут адресоваться индексом параметра, $b(k)$.
<i>NR_PAR_BANDS</i> $\begin{matrix} 10 \\ 20 \\ 34 \end{matrix}$, 10 или 20 полос стерео , 34 полосы стерео
<i>NR_BANDS</i>	Число полос частот, которые могут адресоваться индексом поддиапазона <i>sub</i> , k .

NR_PAR_BANDS	$\begin{matrix} 10 \\ 20 \\ 34 \end{matrix}$, 10 или 20 полос стерео , 34 полосы стерео
$DECAY_CUTOFF$		Стартовая полоса частот для наклона затухания всечастотного фильтра
$DECAY_CUTOFF$	$\begin{matrix} 10 \\ 20 \\ 34 \end{matrix}$, 10 или 20 полос стерео , 34 полосы стерео
$NR_ALLPASS_BANDS$		Число полос всечастотного фильтра
$NR_ALLPASS_BANDS$	$\begin{matrix} 10 \\ 20 \\ 34 \end{matrix}$, 10 или 20 полос стерео , 34 полосы стерео
$SHORT_DELAY_BAND$		Первая полоса стерео, использующая короткую, в одну выборку, задержку.
$SHORT_DELAY_BAND$	$\begin{matrix} 10 \\ 20 \\ 34 \end{matrix}$, 10 или 20 полос стерео , 34 полосы стерео
$a_{smooth} = 0,25$		Коэффициент сглаживания

6.4.5.2 Вычисление декоррелируемого сигнала $d_k(z)$

Процесс декорреляции для первой полосы частот $NR_ALLPASS_BANDS$ $s_k(n)$ основан на всечастотном фильтре, описанном в Z -домене. Его функция передачи для каждой полосы k определяется:

$$H_k(z) = z^2 \Psi_{Fract}(k) \Psi_{NR_ALLPASS_LINKS} \prod_{m=0}^{NR_ALLPASS_LINKS-1} \frac{Q_{Fract_allpass}(k,m)z^{d(m)} - a(m)g_{DecaySlope}(k)}{1 - a(m)g_{DecaySlope}(k)Q_{Fract_allpass}(k,m)z^{d(m)}}$$

для $0 \leq k < NR_ALLPASS_BANDS$.

Матрица длины фрагментарной задержки, $Q_{Fract_allpass}(k,m)$ и вектор фрагментарной задержки, $j_{Fract}(k)$ определяются выражением,

$$Q_{Fract_allpass}(k,m) = \exp(-ipq(m)f_{center}(k)), \begin{matrix} 0 \leq k < NR_ALLPASS_BANDS \\ 0 \leq m < NUM_OF_LINKS \end{matrix},$$

и

$$j_{Fract}(k) = \exp(-ipq_i f_{center}(k)), \quad 0 \leq k < NR_ALLPASS_BANDS, \quad \text{где } i = \sqrt{-1} \text{ обозначает мнимую единицу. Постоянная длины фракционной задержки } q_i = 0,39$$

Вектора коэффициентов фильтра $a(m)$ и вектора длины задержки $d(m)$ см. в таблице 39.

Вектор $g_{DecaySlope}$ содержит независимые от времени факторы для того, чтобы сделать всечастотный фильтр вариантом частотнозависимого. Это определяется выражением:

$$g_{DecaySlope}(k) = \begin{cases} \max(0, 1 - DECAF_SLOPE \cdot (k - DECAF_CUTOFF)) & \text{для } 0 \leq k < NR_ALLPASS_BANDS. \\ 0 & \end{cases}$$

Для верхних полос $NR_ALLPASS_BANDS \leq k < NR_BANDS$ функция передачи $H_k(z)$ равняется задержке согласно:

$$H_k(z) = z^{-D(k)}, \text{ где } D(k) \text{ определяется}$$

$$D(k) = \begin{cases} 4 & , NR_ALLPASS_BANDS \leq k < SHORT_DELAY_BAND \\ 0 & , SHORT_DELAY_BAND \leq k < NR_BANDS \end{cases}$$

6.4.5.3 Обнаружение переходов

Чтобы обработать переходные процессы и другие быстрые огибающие времени, всечастотный фильтр должен быть ослаблен при этих сигналах. Это делается по следующей схеме:

сначала определяется матрица входной мощности $P(i, n)$ которая содержит сумму выборок поддиапазона sub в квадрате каждой полосы параметра

$$P(i, n) = \sum_{s=b(k)}^e |s_k(n)|^2, \quad 0 \leq i < NR_PAR_BANDS,$$

где $b(k)$ определяется в таблице 48 и таблице 49.

Применяется пиковое затухание к мощности входного сигнала согласно:

$$P_{PeakDecayNrg}(i, n) = \begin{cases} P(i, n) & , \alpha P_{PeakDecayNrg}(i, n-1) < P(i, n) \\ \alpha P_{PeakDecayNrg}(i, n-1) & , \text{ в других случаях} \end{cases}$$

для $0 \leq i < NR_PAR_BAND$. α является фактором пикового затухания.

Впоследствии входная мощность и сигналы пикового затухания мощности фильтруются с функцией передачи Z-домена

$$H_{Smooth}(z):$$

$$P_{SmoothNrg}(i, z) = H_{Smooth}(z) \cdot P(i, z),$$

$$P_{SmoothPeakDecayDiffNrg}(i, z) = H_{Smooth}(z) \cdot P_{PeakDecayNrg}(i, z) - P(i, z),$$

для $0 \leq i < NR_PAR_BANDS$, где

$$H_{Smooth}(z) = \frac{a_{Smooth}}{1 + (a_{Smooth} - 1)z^{-1}}$$

Аттенюатор переходов $G_{TransientRatio}$ вычисляется следующим образом:

$$G_{TransientRatio}(i, n) = \begin{cases} \frac{P_{SmoothNrx}(i, n)}{g \Psi_{SmoothPeakDecayDiffNrx}(i, n)} & , g \Psi_{SmoothPeakDecayDiffNrx}(i, n) > P_{SmoothNrx}(i, n) \\ 1 & , \text{ в других случаях} \end{cases}$$

для $0 \leq i < NR_PAR_BANDS$, где $\gamma = 1,5$ фактор воздействия перехода.

Наконец аттенюатор переходов $G_{TransientRatio}$ отображается в полосы согласно:

$$G_{TransientRatioMapped}(k, n) = G_{TransientRatio}(b(k), n) \quad , 0 \leq k < NR_PAR_BANDS.$$

6.4.5.4 Примените сокращения для перехода к декоррелируемому сигналу

Пусть $d_k(z)$ будет декоррелируемым сигналом и $s_k(z)$ входным сигналом моно в Z-домене для каждой полосы. Тогда $d_k(z)$ определяется согласно:

$$d_k(z) = G_{TransientRatioMapped}(k, z) \Psi_k(z) \Psi_k(z), \text{ где } 0 \leq k < NR_PAR_BANDS.$$

Таблица 39 - Векторы коэффициентов фильтра, векторы длины задержки $d_{24kHz}(m)$ и $d_{48kHz}(m)$

m	$a(m)$	$d(m)$
0	0,65143905753106	3
1	0,56471812200776	4
2	0,48954165955695	5

Таблица 40 - Вектор длины задержки f_{center_20}

k	$f_{center_20}(k)$	k	$f_{center_20}(k)$
0	-3/8	5	7/8
1	-1/8	6	5/4
2	1/8	7	7/4
3	3/8	8	9/4
4	5/8	9	11/4

Таблица 41 - Вектор длины задержки f_{center_34}

k	$f_{center_34}(k)$	k	$f_{center_34}(k)$
0	1/12	16	9/8
1	3/12	17	11/8
2	5/12	18	13/8
3	7/12	19	15/8
4	9/12	20	9/4
5	11/12	21	11/4
6	13/12	22	13/4
7	15/12	23	7/4
8	17/12	24	17/4
9	-5/12	25	11/4
10	-3/12	26	13/4
11	-1/12	27	15/4
12	17/8	28	17/4
13	19/8	29	19/4
14	5/8	30	21/4
15	7/8	31	15/4

$$f_{center_20}(k) = k + \frac{1}{2} - 7, \quad , 10 \leq k < NR_ALLPASS_BANDS.$$

$$f_{center_34}(k) = k + \frac{1}{2} - 27, \quad , 32 \leq k < NR_ALLPASS_BANDS.$$

$$f_{center} \begin{cases} Mf_{center_20} \\ 0f_{center_34} \end{cases}, \quad \begin{matrix} , NR_PAR_BANDS = 20 \\ , NR_PAR_BANDS = 34 \end{matrix}$$

Таблица 42 - Фракционный вектор длины задержки $q(m)$

(m)	$q(m)$
0	0,43
1	0,75
2	0,347

Таблица 43 - Пиковые факторы затухания α

α	0,76592833836465
----------	------------------

6.4.6 Обработка стерео

Наборы выборок поддиапазона $sub\ s_k(n)$ и $d_k(n)$ обрабатываются согласно сигналам стерео. Эти сигналы определяются на полосу стерео. Все гибридные выборки поддиапазона в пределах полосы стерео обрабатываются согласно сигналам в этой соответствующей полосе стерео. Таблица 48 и таблица 49 указывают гибридные выборки поддиапазона, которые попадают в каждый стереодиапазон для конфигурации стереодиапазона (10,20) и 34. k проходит диапазон от $[0 \dots 70]$ или $[0 \dots 90]$ для конфигурации стереодиапазона (10,20) или 34, соответственно.

6.4.6.1 Отображение

Число полос стерео, которое фактически используется для обработки сигналов, зависит от числа IID и доступных параметров для ICC согласно отношению, данному в таблице 44. В случае, если никакие IID или параметры ICC не были переданы в текущем фрейме ($enable_iid = \%0$ или $enable_icc = \%0$), число IID или параметр ICC , соответственно, считается равным 20. В случае, если никакой IID и никакие параметры ICC не были переданы в текущем фрейме ($enable_iid = \%0$ и $enable_icc = \%0$), число полос стерео в предыдущем фрейме сохраняется неизменным и используется также для обработки текущего фрейма.

Таблица 44 - Число полос стерео зависит от числа параметров для IID и ICC

Число параметров IID	Число параметров ICC	Число полос стерео
10	10	20 (то есть, конфигурация 10,20 полос стерео)
10	20	
20	10	
20	20	
10,20	34	34
34	10, 20	
34	34	

В случае, когда число параметров для IID и ICC отличается от числа полос стерео, требуется отображение из меньшего числа параметров в большее число параметров. Для отображения от 10 до 20 параметров это реализуется копированием каждого параметра как показано в таблице 45. Для отображения от 20 до 34 параметров это реализуется согласно таблице 45. Для отображения от 10 до 34 параметров сначала 10 параметров отображаются на 20 параметров, а потом на 34 параметра. Таблица 46 дает инверсное отображение от 34 параметров до 20.

Таблица 45 – Отображение от 10 до 20 параметров в 34

Сетка параметров			Сетка параметров		
34	20	10	34	20	10
idx_0	idx_0	idx_0	idx_{17}	idx_{11}	idx_5
idx_1	$(idx_0+idx_1)/2$	idx_0	idx_{18}	idx_{12}	idx_6
idx_2	idx_1	idx_0	idx_{19}	idx_{13}	idx_6
idx_3	idx_2	idx_1	idx_{20}	idx_{14}	idx_7
idx_4	$(idx_2+idx_3)/2$	idx_1	idx_{21}	idx_{14}	idx_7
idx_5	idx_3	idx_1	idx_{22}	idx_{15}	idx_7
idx_6	idx_4	idx_2	idx_{23}	idx_{15}	idx_7
idx_7	idx_4	idx_2	idx_{24}	idx_{16}	idx_8
idx_8	idx_5	idx_2	idx_{25}	idx_{16}	idx_8
idx_9	idx_5	idx_2	idx_{26}	idx_{17}	idx_8
idx_{10}	idx_6	idx_3	idx_{27}	idx_{17}	idx_8
idx_{11}	idx_7	idx_3	idx_{28}	idx_{18}	idx_9
idx_{12}	idx_8	idx_4	idx_{29}	idx_{18}	idx_9
idx_{13}	idx_8	idx_4	idx_{30}	idx_{18}	idx_9
idx_{14}	idx_9	idx_4	idx_{31}	idx_{18}	idx_9
idx_{15}	idx_9	idx_4	idx_{32}	idx_{19}	idx_9
idx_{16}	idx_{10}	idx_5	idx_{33}	idx_{19}	idx_9

Таблица 46 - Отображение параметров *IID*, *ICC*, *IPD* и *OPD* из 34 полос стерео в 20 полос стерео. Для параметров *IPD* и *OPD* это отображение применяется до и включая idx_{10} и idx_{16} для 20 и 34 полос стерео соответственно

20 полос стерео	34 полосы стерео
idx_0	$(2*idx_0 + idx_1)/3$
idx_1	$(idx_1 + 2*idx_2)/3$
idx_2	$(2*idx_3 + idx_4)/3$
idx_3	$(idx_4 + 2*idx_5)/3$
idx_4	$(idx_6 + idx_7)/2$
idx_5	$(idx_8 + idx_9)/2$
idx_6	idx_{10}
idx_7	idx_{11}
idx_8	$(idx_{12} + idx_{13})/2$
idx_9	$(idx_{14} + idx_{15})/2$
idx_{10}	idx_{16}
idx_{11}	idx_{17}
idx_{12}	idx_{18}
idx_{13}	idx_{19}
idx_{14}	$(idx_{20} + idx_{21})/2$
idx_{15}	$(idx_{22} + idx_{23})/2$
idx_{16}	$(idx_{24} + idx_{25})/2$
idx_{17}	$(idx_{26} + idx_{27})/2$
idx_{18}	$(idx_{28} + idx_{29} + idx_{30} + idx_{31})/4$
idx_{19}	$(idx_{32} + idx_{33})/2$

Процесс усреднения, обозначенный $(idx_0 + idx_1)/2$ в таблице 45 и $(2*idx_0 + idx_1)/2$ в таблице 46, выполняется для представления целочисленного индекса idx_k параметров *IID* или *ICC* перед деквантизацией, согласно целочисленной арифметике *ANSI-C*.

Параметры *IPD/OPD* следуют за отображением для параметров *IID*, принимая во внимание относительное количество параметров для *IPD/OPD*. Следовательно для *IPD/OPD* применяется то же самое отображение как для *IID*, но только для более низкого числа параметров. Для верхних полос стерео, где никакие данные *IPD/OPD* не передаются, параметры *IPD/OPD* обнуляются.

Если число полос стерео изменяется от 10, 20 в предыдущем фрейме к 34, коэффициенты $h_{11}(b)$, $h_{12}(b)$, $h_{21}(b)$, и $h_{22}(b)$ в конце предыдущего фрейма отображаются от 20 до 34 полос стерео согласно таблице 45 (путем замены idx_b на $h_{ij}(b)$, где ij равно 11, 12, 21, или 22). Коэффициенты $H_{11}(k,n)$, $H_{12}(k,n)$, $H_{21}(k,n)$, и $H_{22}(k,n)$ для конца предыдущего фрейма извлекаются согласно четырем уравнениям, приведенным в 6.4.6.3 перед дальнейшей обработкой, как определено в 6.4.6.4. Параметры состояния сглаживания IPD/OPD сбрасываются, то есть $opd(b, n_{e-1})=0$, $ipd(b, n_{e-1})=0$, $opd(b, n_e)=0$ и $ipd(b, n_e)=0$. Разрешающая способность по частоте гибридного блока фильтров анализа QMF изменяется на конфигурацию 34 полосы стерео. Параметры состояния процесса декорреляции сбрасываются в нуль.

Если число полос стерео изменяется от 34 в предыдущем фрейме к 10,20 в текущем фрейме, коэффициенты $h_{11}(b)$, $h_{12}(b)$, $h_{21}(b)$ и $h_{22}(b)$ в конце предыдущего фрейма отображаются от 34 до 20 полос стерео согласно таблице 46 (заменяя idx_b на $h_{ij}(b)$, где ij равно 11, 12, 21, или 22). Коэффициенты $H_{11}(k,n)$, $H_{12}(k,n)$, $H_{21}(k,n)$ и $H_{22}(k,n)$ для конца предыдущего фрейма извлекаются согласно четырем уравнениям, приведенным в 6.4.6.3 перед дальнейшей обработкой, как определено в 6.4.6.4. Переменные состояния сглаживания IPD/OPD сбрасываются, то есть, $opd(b, n_{e-1})=0$, $ipd(b, n_{e-1})=0$, $opd(b, n_e)=0$ и $ipd(b, n_e)=0$. Разрешающая способность по частоте гибридного блока фильтров анализа QMF изменяется на конфигурацию 20 полос стерео. Параметры состояния процесса декорреляции сбрасываются, чтобы обнулиться (таблица 47).

Таблица 47 – Изменение числа полос стерео.

Предыдущий фрейм	Текущий фрейм	
	10/20 полос	34 полосы
10/20 полос	-	Отображение $h_{ij}(b)$ согласно таблицы 45, сброс параметров состояния
34 полосы	Отображение $h_{ij}(b)$ согласно таблицы 46, сброс параметров состояния	-

6.4.6.2 Смешивание

Чтобы генерировать сигналы поддиапазона QMF для выборок поддиапазона $n = n_e + 1 \dots n_{e+1}$ требуются параметры в положении n_e и n_{e+1} , а также сигналы как домена поддиапазона $s_{\lambda}(n)$ и $d_{\lambda}(n)$ для $n = n_e + 1 \dots n_{e+1}$. Для IPD/OPD дополнительно не-

обходимы параметры в позиции n_{e-1} . n_e представляет стартовую позицию для огибающей e . В случае $e=0$, n_{-1} представляет позицию последнего параметра предыдущего фрейма стерео. Для первого фрейма стерео параметры IPD/OPD в позиции n_{-1} инициализируются в нуль. В случае класса фрейма == %1 ($VAR_BORDERS$), позиции границы n_e получаются с помощью $border_position [e]$. В случае класса фрейма == %0 ($FIX_BORDERS$), позиции границы n_e получаются посредством формулы

$$n_e = \left\lfloor \frac{numQMFSlots * (e + 1)}{num_env} \right\rfloor + 1, \quad e=[0, \dots, num_env-1]$$

Сигналы поддиапазона sub стерео создаются как:

$$l_k(n) = H_{11}(k, n)s_k(n) + H_{21}(k, n)d_k(n)$$

$$r_k(n) = H_{11}(k, n)s_k(n) + H_{21}(k, n)d_k(n)$$

Чтобы получить матрицы $H_{11}(k, n)$, $H_{12}(k, n)$, $H_{21}(k, n)$, и $H_{22}(k, n)$ сначала должны быть вычислены векторы $h_{11}(b)$, $h_{12}(b)$, $h_{21}(b)$, $h_{22}(b)$, где параметр b используется в качестве индекса параметра. Сначала преобразуются к линейному домену различия в интенсивности (IID) для позиции параметра n_{e+1}

$$c(b) = 10^{\frac{iid(b)}{20}},$$

где $iid(b)$ представляет декодированное значение IID для полосы стерео b в дБ. В зависимости от конфигурации режима ICC используются смешивающая процедура R_a или R_b . Для обеих смешивающих процедур используются параметры для позиции параметра n_{e+1} .

6.4.6.2.1 Смешивающая процедура R_a

В случае использования смешивающей процедуры R_a применяется следующий метод.

Из различий в интенсивности вычисляются два вектора масштабного коэффициента c_1 и c_2

$$c_1(b) = \frac{\sqrt{2}}{\sqrt{1 + c^2(b)}}$$

$$c_2(b) = \frac{\sqrt{2c(b)}}{\sqrt{1 + c^2(b)}}$$

Исходя из них и параметра $ICC \rho(b)$ вычисляются коэффициенты $h_{xy}(b)$ согласно

$$a(b) = \frac{1}{2} \arccos(r(b))$$

$$b(b) = a(b) \frac{c_1(b) - c_2(b)}{\sqrt{2}}$$

$$h_{11}(b) = \cos(a(b) + b(b))c_2(b)$$

$$h_{12}(b) = \cos(b(b) - a(b))c_1(b)$$

$$h_{21}(b) = \sin(a(b) + b(b))c_2(b)$$

$$h_{22}(b) = \sin(b(b) - a(b))c_1(b)$$

6.4.6.2.2 Процедура смешивания R_b

В случае, когда используется процедура смешивания R_b , применяется следующий метод.

Чтобы предотвратить неустойчивость, в случае значения $\rho(b)$ меньше чем 0,05, $\rho(b)$ устанавливается в 0,05. В случае если $c(b)$ не равно 1

$$a(b) = \frac{1}{2} \arctan \frac{2c(b)r(b)}{c^2(b) - 1}$$

иначе $a(b) = \frac{\rho}{4}$. После исправления по модулю $a(b)$ значения $c(b)$ и $\rho(b)$ снова используются, чтобы получить коэффициенты $h_{ij}(b)$.

$$a(b) = a(b) - \frac{1}{2} \arctan \frac{2c(b)r(b)}{c^2(b) - 1} \rho,$$

$$m(b) = 1 + \frac{4r^2(b) - 4}{(c(b) + c^{-1}(b))^2},$$

$$g(b) = \arctan \left(\sqrt{\frac{1 - \sqrt{m(b)}}{1 + \sqrt{m(b)}}} \right),$$

$$h_{11}(b) = \sqrt{2} \cos(a(b)) \cos(g(b)),$$

$$h_{12}(b) = \sqrt{2} \sin(a(b)) \cos(g(b)),$$

$$h_{21}(b) = -\sqrt{2} \sin(a(b)) \sin(g(b)),$$

$$h_{22}(b) = \sqrt{2} \cos(a(b)) \sin(g(b)).$$

6.4.6.3 Фазовые параметры

6.4.6.3.1 Фазовые параметры заблокированы

Если *IPD* и *OPD* заблокированы ($enable_ipdopd == 0$), применяется следующая процедура. Чтобы получить $H_{11}(k, n_{e+1})$, $H_{12}(k, n_{e+1})$, $H_{21}(k, n_{e+1})$ и $H_{22}(k, n_{e+1})$, используем следующие уравнения:

$$H_{11}(k, n_{e+1}) = h_{11}(b(k))$$

$$H_{12}(k, n_{e+1}) = h_{12}(b(k))$$

$$H_{21}(k, n_{e+1}) = h_{21}(b(k))$$

$$H_{22}(k, n_{e+1}) = h_{22}(b(k))$$

где $b(k)$ определяется в таблице 48 и таблице 49.

6.4.6.3.2 Фазовые параметры задействованы

В случае, если *IPD* и *OPD* задействованы ($enable_ipdopd == 1$), применяется следующая процедура. Сначала значения *IPD* и *OPD* сглаживаются во времени согласно

$$j_{opd}(b) = \left(\frac{M}{84} \exp(j\psi_{pd}(b, e-1)) + \frac{1}{2} \exp(j\psi_{pd}(b, e)) + \exp(j\psi_{pd}(b, e+1)) \right) \frac{b}{B}$$

$$j_{ipd}(b) = \left(\frac{M}{84} \exp(j\psi_{pd}(b, e-1)) + \frac{1}{2} \exp(j\psi_{pd}(b, e)) + \exp(j\psi_{pd}(b, e+1)) \right) \frac{b}{B}$$

В случае, когда число параметров *IPD/OPD* для позиции параметра n_{e-1} и/или n_e отличается от числа параметров *IPD/OPD* для позиции параметра n_{e+1} , они отображаются в число параметров *IPD/OPD* для позиции параметра n_{e+1} , используя таблицу 45 и таблицу 46.

$$j_1(b) = j_{opd}(b)$$

$$j_2(b) = j_{ipd}(b) - j_{opd}(b)$$

Векторы $h_{11}(b)$, $h_{12}(b)$, $h_{21}(b)$, $h_{22}(b)$ изменяются согласно:

$$h_{11}(b) = h_{11}(b) \cdot \exp(j\varphi_1(b))$$

$$h_{12}(b) = h_{12}(b) \cdot \exp(j\varphi_2(b))$$

$$h_{21}(b) = h_{21}(b) \cdot \exp(j\varphi_1(b))$$

$$h(b) = h_{22}(b) \cdot \exp(j\varphi_2(b))$$

Чтобы получить $H_{11}(k, n_{e+1})$, $H_{12}(k, n_{e+1})$, $H_{21}(k, n_{e+1})$ и $H_{22}(k, n_{e+1})$ используются следующие уравнения

$$H_{11}(k, n_{e+1}) = h_{11}(b(k))$$

$$H_{12}(k, n_{e+1}) = h_{12}(b(k))$$

$$H_{21}(k, n_{e+1}) = h_{21}(b(k))$$

$$H_{22}(k, n_{e+1}) = h_{22}(b(k))$$

где $b(k)$ определяется в таблице 48 и таблице 49. Для индексов, обозначенных a^* используются следующие уравнения

$$H_{11}(k, n_{e+1}) = h_{11}^*(b(k))$$

$$H_{12}(k, n_{e+1}) = h_{12}^*(b(k))$$

$$H_{21}(k, n_{e+1}) = h_{21}^*(b(k))$$

$$H_{22}(k, n_{e+1}) = h_{22}^*(b(k)),$$

где h_{ii}^* представляет комплексно сопряженным с h_{ii} .

Таблица 48 - Отображение параметров от 20 полос до 71 поддиапазона

Индекс k поддиапазона	Канал QMF	Индекс параметра $b(k)$	Примечание
0	0	1*	<i>Sub QMF</i>
1	0	0*	
2	0	0	
3	0	1	
4	0	2	
5	0	3	
6	1	4	
7	1	5	
8	2	6	
9	2	7	
10	3	8	
11	4	9	
12	5	10	
13	6	11	
14	7	12	
15	8	13	
16-17	9-10	14	
18-20	11-13	15	
21-24	14-17	16	
25-29	18-22	17	
30-41	23-34	18	
42-70	35-63	19	

Таблица 49 - Отображение параметров из 34 полос в 91 поддиапазон *sub*

Индекс <i>k</i> поддиапазона	Канал <i>QMF</i>	Индекс параметра <i>b</i> (<i>k</i>)	Примечание
0	0	0	Sub QMF
1	0	1	
2	0	2	
3	0	3	
4	0	4	
5	0	5	
6-7	0	6	
8	0	7	
9	0	2*	
10	0	1*	
11	0	0*	
12-13	1	10	
14	1	4	
15	1	5	
16	1	6	
17	1	7	
18	1	8	
19	1	9	
20	2	10	
21	2	11	
22	2	12	
23	2	9	
24	3	14	
25	3	11	
26	3	12	
27	3	13	
28	4	14	
29	4	15	
30	4	16	
31	4	13	
32	5	16	QMF (только)
33	6	17	

Окончание таблицы 49			
Индекс k поддиапазона	Канал QMF	Индекс параметра $b(k)$	Примечание
34	7	18	
35	8	19	
36	9	20	
37	10	21	
38-39	11-12	22	
40-41	13-14	23	
42-43	15-16	24	
44-45	17-18	25	
46-47	19-20	26	
48-50	21-23	27	
51-53	24-26	28	
54-56	27-29	29	
57-59	30-32	30	
60-63	33-36	31	
64-67	37-40	32	
68-90	41-63	33	

6.4.6.4 Интерполяция

Промежуточные значения для $H_{11}(k, n)$, $H_{12}(k, n)$, $H_{21}(k, n)$ и $H_{22}(k, n)$ в позициях $n = n_e + 1 \dots n_{e+1}$ получаются посредством линейной интерполяции, соответствующей

$$H_{11}(k, n) = H_{11}(k, n_e) + (n - n_e) \frac{H_{11}(k, n_{e+1}) - H_{11}(k, n_e)}{n_{e+1} - n_e}$$

$$H_{12}(k, n) = H_{12}(k, n_e) + (n - n_e) \frac{H_{12}(k, n_{e+1}) - H_{12}(k, n_e)}{n_{e+1} - n_e}$$

$$H_{21}(k, n) = H_{21}(k, n_e) + (n - n_e) \frac{H_{21}(k, n_{e+1}) - H_{21}(k, n_e)}{n_{e+1} - n_e}$$

$$H_{22}(k, n) = H_{22}(k, n_e) + (n - n_e) \frac{H_{22}(k, n_{e+1}) - H_{22}(k, n_e)}{n_{e+1} - n_e}$$

Особые случаи:

а) для первой области фрейма стерео с $n = 0 \dots n_0 - 1$ применяется следующее:

$$H_{11}(k, n) = H_{11}(k, n_{-1}) + n \frac{H_{11}(k, n_0) - H_{11}(k, n_{-1})}{n_0}$$

$$H_{12}(k, n) = H_{12}(k, n_{-1}) + n \frac{H_{12}(k, n_0) - H_{12}(k, n_{-1})}{n_0}$$

$$H_{21}(k, n) = H_{21}(k, n_{-1}) + n \frac{H_{21}(k, n_0) - H_{21}(k, n_{-1})}{n_0}$$

$$H_{22}(k, n) = H_{22}(k, n_{-1}) + n \frac{H_{22}(k, n_0) - H_{22}(k, n_{-1})}{n_0}$$

где $H_{xx}(k, n_{-1})$ представляет коэффициенты $H_{xx}(k, n_{num_env-1})$, полученные из предыдущего фрейма стерео.

Для первого фрейма стерео $H_{xx}(k, n_{-1})$ инициализируется в нуль,

б) для последней области фрейма стерео с $n = (k, n_{num_env-1}) \dots numQMFSlots-1$ применяется следующее:

$$H_{11}(k, n) = H_{11}(k, n_{num_env-1})$$

$$H_{12}(k, n) = H_{12}(k, n_{num_env-1})$$

$$H_{21}(k, n) = H_{21}(k, n_{num_env-1})$$

$$H_{22}(k, n) = H_{22}(k, n_{num_env-1})$$

6.4.6.5 Процедура для неполных наборов параметров

В случае, если никакие параметры не были переданы в текущем фрейме ни для *IID*, *ICC*, ни для *IPD/OPD* или их комбинации, значения параметров для текущего фрейма получаются согласно переменной *num_env*, как дается в таблицах 50, 51 и 52.

Таблица 50 - Источник параметров для *IID* в случае отсутствия передачи параметров

	<i>enable_iid</i>	
	0	1
<i>num_env</i> =0	Набор параметров <i>IID</i> , чтобы присвоить значение по умолчанию	Сохраненные параметры <i>IID</i>
<i>num_env</i> > 0	Набор параметров <i>IID</i> , чтобы присвоить значение по умолчанию	Не определено

Таблица 51 - Источник параметров для *ICC* в случае отсутствия передачи параметров

	<i>enable_icc</i>	
	0	1
<i>num_env</i> =0	Набор параметров <i>ICC</i> , чтобы присвоить значение по умолчанию	Сохраненные параметры <i>ICC</i>
<i>num_env</i> >0	Набор параметров <i>ICC</i> , чтобы присвоить значение по умолчанию	Не определено

Таблица 52 - Источник параметров для *IPD/OPD* в случае отсутствия передачи параметров

	<i>enable_ipdopd</i>	
	0	1
<i>num_env</i> =0	Набор параметров <i>IPD/OPD</i> , чтобы присвоить значение по умолчанию	Сохраненные параметры <i>IPD/OPD</i>
<i>num_env</i> >0	Набор параметров <i>IPD/OPD</i> , чтобы присвоить значение по умолчанию	Не определено

В случае, если параметры должны быть установлены в значение по умолчанию, параметры в позициях, определенных n_e устанавливаются в их значения по умолчанию (*index*=0).

В случае, если параметры должны быть сохранены, различают две ситуации. Если *enable_ipdopd* == %1, четыре вектора $H_{11}(k, n)$, $H_{12}(k, n)$, $H_{21}(k, n)$ и $H_{22}(k, n)$ для всего $n = [0, \dots, \text{numQMFSlots}-1]$ копируются из те же самых четырех векторов в позиции $n=\text{numQMFSlots}-1$ в предыдущем элементе *ps_data* (). Если *enable_ipdopd* == %0, четыре вектора $H_{11}(k, n)$, $H_{12}(k, n)$, $H_{21}(k, n)$ и $H_{22}(k, n)$ для всего $n = [0, \dots, \text{numQMFSlots}-1]$ устанавливаются в четыре вектора $h_{11}(b(k))$, $h_{12}(b(k))$, $h_{21}(b(k))$ и $h_{22}(b(k))$, соответственно, где $n=\text{numQMFSlots}-1$ в предыдущем элементе *ps_data* ().

6.4.7 Гибридный блок фильтров синтеза *QMF*

Гибридные сигналы поддиапазона обработки стерео $l_k(n)$ и $r_k(n)$ подаются в гибридные блоки фильтров синтеза, которые реализуются как сумматоры выборок *sub QMF*. Два блока фильтров синтеза идентичны блоку фильтров синтеза 64 комплексных *QMF*. Вводом в блок фильтров являются слоты 64 выборки *QMF*. Для каждого слота блок фильтров выводит один блок из 64 выборок одного канала реконструируемого сигнала стерео.

6.5 Ситуации запуска/остановки для декодирования

Декодирование выборки должно запускаться и заканчиваться определенным образом. Этот подпункт объясняет, как поступать с запуском и остановкой процесса декодирования.

6.5.1 Запуск декодирования

Запуск декодирования происходит для первого фрейма выборки или во время произвольного доступа в выборке.

Для синусоид предыдущий (несуществующий) подфрейм должен быть заполнен сигналом нуля. Тогда методом перекрытия-добавления генерируется естественное постепенное появление синусоидальных компонентов первого подфрейма.

Для шума предыдущий (несуществующий) подфрейм должен быть заполнен сигналом нуля. Метод перекрытия-добавления тогда генерирует естественное постепенное появление шумового компонента первого подфрейма.

Совместимый декодер, который получает данные *PS*, должен вывести моносигнал в два выходных канала, пока не будет принят первый элемент *ps_data ()* с *enable_ps_header == 1*, в котором для всех задействованных параметров используется дифференциальное кодирование частоты и *num_env > 0*, гарантируя, что данные *PS* могут правильно декодироваться.

6.5.2 Прекращение декодирования

Прекращение декодирования происходит для последнего фрейма выборки, или во время произвольного доступа в выборке (прекращение процесса декодирования "вручную" (например, остановка, пропуск, пауза)).

Для шага переходного процесса не требуются никакие специальные предосторожности. Для переходного процесса Мейкслера возможно, что хвостовая часть не закончилась в конце выборки. Рекомендуется прекратить генерацию вывода для переходного процесса Мейкслера в конце выборки.

Для синусоид следующий (несуществующий) подфрейм должен быть заполнен сигналом нуля. Метод перекрытия-добавления тогда генерирует естественное постепенное исчезновение для синусоидальных компонентов последнего подфрейма.

Для шума следующий (несуществующий) подфрейм должен быть заполнен сигналом нуля. Метод перекрытия-добавления используется когда генерирует естественное постепенное исчезновение для шумового компонента последнего подфрейма.

Приложение А

(обязательное)

Комбинация инструмента *SBR* с параметрическим инструментом стерео

А.1 Краткий обзор

Параметрический инструмент кодирования стерео (инструмент *PS*) может использоваться в комбинации с инструментом *SBR*. В этом случае аудиосигнал с первого канала передается в *AAC+SBR* (то есть, *HE-AAC*), и инструмент *PS* используется, чтобы восстановить сигнал второго стерео канала от этого монофонического сигнала. Элемент потока битов *ps_data ()* передает информацию, необходимую инструменту *PS*, и переносится в *sbr_extension ()* контейнер потока битов *SBR*.

Использование этого параметрического расширения стерео *HE AAC* сообщается или неявно присутствием параметрических данных стерео в потоке битов, или явно сигнализируя соответствующий *AudioObjectType* в *audioSpecificConfig*. Следовательно, неявная сигнализация требует, если *sbr_extension ()* с *bs_extension_id == EXTENSION_ID_PS* находится в части *SBR* потока битов, декодер, поддерживающий комбинацию *SBR* и *PS*, должен управлять инструментом *PS*, чтобы генерировать выходной сигнал стерео. Если никакой элемент *ps_data ()* не доступен в части *SBR* монофонического потока битов *HE-AAC*, нормальный монофонический сигнал сгенерирован инструментом *SBR* и отображен на выходной сигнал стерео, в котором левый и правый канал содержат тот же самый монофонический сигнал.

А.2 Синтаксис потока битов и семантика

Элемент потока битов *ps_data ()* переносят в *sbr_extension ()* контейнер, обеспеченный потоком битов *SBR*.

Таблица А. 1 - Синтаксис *sbr_extension ()*

Синтаксис	Количество битов	Мнемосхема
<pre> sbr_extension (bs_extension_id, num_bits_left) { switch (bs_extension_id) { case EXTENSION_ID_PS: num_bits_left = ps_data (); break; default: </pre>		Примечание

Окончание таблицы А.1		
Синтаксис	Количество битов	Мнемосхема
<pre> bs_fill_bits; num_bits_left = 0; break; } } </pre>	<i>num_bits_left</i>	<i>bslbf</i>
Примечание : <i>ps_data ()</i> возвращает число чтения битов.		

А.3 Процесс декодирования

Семантика и декодирующий процесс для инструмента *PS* определяется в 5.2 и 6.4. Когда инструмент *PS* объединяется с *SBR*, фрейм стерео идентичен фрейму *SBR* и состоит из 32 сложных выборок на полосу *QMF* для 1024 структурированной AAC (30 выборок для 960 структурированной).

Чтобы позволить эффективную реализацию инструмента *PS*, частичный сброс *decorrelator* параметров состояния выполняется для каждого фрейма стерео для всех поддиапазонов *QMF* выше самого высокого поддиапазона *QMF*, сгенерированного инструментом *SBR*

$$d_k(n)=0$$

$$s_k(n)=0,$$

где $n < n_e$, $k_{max} \leq k < NR_BANDS$, n_e является первой выборкой в текущем фрейме стерео и

$$k_{max} = k_e + M + \begin{cases} 10 & , 10 \text{ или } 20 \text{ полос стерео} \\ 27 & , 34 \text{ полосы стерео} \end{cases}$$

Если никакой *ps_data ()* элемент не присутствовал в предыдущем фрейме, то полный сброс *decorrelator* параметров состояния выполняется, вызывая состояния :

$$d_k(n)=0$$

$$s_k(n)=0,$$

где $n < n_e$, $0 \leq k < NR_BANDS$, и n_e является первой выборкой в текущем фрейме стерео.

Инструмент *PS* использует оцененное комплексное представление *QMF*, и поэтому не может использоваться в комбинации с низкой версией питания инструмента *SBR*. Если *DRC* используется в комбинации с *SBR*, то *DRC* применяется в домене *QMF* к выводу инструмента *PS* до синтеза *QMF filterbanks*. Тот же самый *factor(k, l)* применяется и к левому и к правому звуковому каналу.

А.4 Базовая версия параметрического инструмента кодирования стерео

Чтобы облегчить реализацию инструмента декодера *PS* на платформах с очень ограниченными вычислительными ресурсами, используется базовая версия инструмента *PS*. Декодер *PS*, реализовывая эту базовую версию, всегда использует гибридную структуру фильтра для 20 полос стерео и не реализует синтез *IPD/OPD* и смешивающийся режим *Rb*. Это приводит к сокращению вычислительной сложности приблизительно на 25 % по сравнению с неограниченным инструментом *PS*. Базовая версия инструмента *PS* поддерживает полный синтаксис потока битов для *ps_data* (). Однако, данные *IPD/OPD* игнорируются и сбрасываются к $IPD=OPD=0$ до синтеза стерео. Если 34 конфигурации полосы стерео используются для *IID* или параметров *ICC* в потоке битов, декодируемые параметры отображаются на 20 полос стерео. Базовый декодер всегда использует смешивающий *Pa* режим независимо от значения *icc_mode*.

Приложение Б

(обязательное)

Б.1 Таблицы Хаффмана для SSC

Функция *ssc_huff_dec* () используется как:

date = *ssc_huff_dec* (*t_huff*, *codeword*),

где *t_huff* является выбранной таблицей Хаффмана, и *codeword* является словом, считанным из потока битов.

Таблица Б.1 - *huff_sgrid*

Индекс	<i>huff_sgrid</i>
0	100001
1	11101
2	11110
3	1100
4	1101
5	1010
6	0111
7	001
8	1011
9	0110
10	1001
11	0101
12	0000
13	0001
14	11100
15	01001
16	111111
17	111110
18	100000
19	010001
20	010000
21	10001

Таблица Б.2 - *huff_sampba*

Индекс	<i>huff_sampba</i>	Индекс	<i>huff_sampba</i>	Индекс	<i>huff_sampba</i>	Индекс	<i>huff_sampba</i>
0	110010010	64	1101	128	0110	192	110010011001
8	0100111	72	001	136	11000	200	110010011000100
16	1100101	80	000	144	01000	208	110010011000101
24	110011	88	1111	152	010010	216	110010011000110
32	01110	96	1110	160	0100110	224	110010011000111
40	01111	104	1011	168	11001000	232	11001001100000
48	0101	112	1010	176	1100100111	240	11001001100001
56	1001	120	1000	184	11001001101		

Таблица Б.3 - *huff_sampbr*

Индекс	<i>huff_sampbr</i>	Индекс	<i>huff_sampbr</i>	Индекс	<i>huff_sampbr</i>
-240	11111110110001000010	-72	111101101	96	111111101100011
-232	11111110110001000011	-64	11110111	104	11111110110101
-224	11111110110001000100	-56	1111010	112	11111110110000
-216	11111110110001000101	-48	111100	120	11111110110001010001
-208	11111110110001000110	-40	111110	128	11111110110001010010
-200	11111110110001000111	-32	11101	136	11111110110001010011
-192	11111110110001001000	-24	0111	144	11111110110001010100
-184	11111110110001001001	-16	010	152	11111110110001010101
-176	11111110110001001010	-8	00	160	11111110110001010110
-168	11111110110001001011	0	10	168	11111110110001010111
-160	11111110110001001100	8	110	176	11111110110001011000
-152	11111110110001001101	16	0110	184	11111110110001011001
-144	11111110110001001110	24	11100	192	11111110110001011010
-136	11111110110001001111	32	111110	200	11111110110001011011
-128	11111110110001010000	40	1111110	208	11111110110001011100
-120	1111111011010100	48	111101100	216	11111110110001011101
-112	111111101101001	56	111111110	224	11111110110001011110
-104	1111111111100	64	1111111010	232	11111110110001011111
-96	11111110111	72	11111111111	240	1111111011000100000
-88	1111111110	80	11111111101		
-80	1111111100	88	1111111011011		

Таблица Б.4 - *huff_sampca*

Индекс	<i>huff_sampca</i>	Индекс	<i>huff_sampca</i>	Индекс	<i>huff_sampca</i>
0	01101101011	88	000	176	0011001
8	01101100	96	1111	184	011011011
16	0011000	104	1110	192	0110110100
24	0110111	112	1100	200	011011010101
32	011010	120	1011	208	0110110101001
40	00111	128	1000	216	01101101010001
48	10011	136	0101	224	0110110101000010
56	0100	144	0010	232	0110110101000011
64	0111	152	10010	240	011011010100000
72	1010	160	01100		
80	1101	168	001101		

Таблица Б.5 - *huff_sampcr* [0]

Индекс	<i>huff_sampcr</i> [0]	Индекс	<i>huff_sampcr</i> [0]	Индекс	<i>huff_sampcr</i> [0]
-26	111001010111000	-8	0110111	10	01101100
-25	01101101110	-7	001000	11	00011010
-24	1110010101111	-6	111000	12	011011010
-23	000111001011	-5	00101	13	000111010
-22	011010111110	-4	11101	14	1110010111
-21	111001010100	-3	0011	15	0110110110
-20	00011100100	-2	1111	16	0001110011
-19	01101011110	-1	110	17	11100101101
-18	11100101100	0	10	18	01101101111
-17	0001110110	1	010	19	00011101110
-16	0110101110	2	0111	20	111001010110
-15	1110010100	3	0000	21	111001010101
-14	000111000	4	01100	22	011010111111
-13	011010110	5	00010	23	000111001010
-12	00011011	6	001001	24	11100101011101
-11	01101010	7	1110011	25	00011101111
-10	11100100	8	0110100	26	111001010111001
-9	0001111	9	0001100		

Таблица Б.6 - *huff_sampr* [1]

Индекс	<i>huff_sampr</i> [1]	Индекс	<i>huff_sampr</i> [1]	Индекс	<i>huff_sampr</i> [1]
-26	100111001011	-8	110110	10	11011111
-24	11011100010	-6	11010	12	10011101
-22	110111101110	-4	1100	14	110111001
-20	11011110110	-2	111	16	1101111010
-18	1101110000	0	0	18	1001110011
-16	100111000	2	101	20	11011100011
-14	110111100	4	1000	22	10011100100
-12	11011101	6	10010	24	110111101111
-10	1001111	8	100110	26	100111001010

Таблица Б.7 - *huff_sampr* [2]

Индекс	<i>huff_sampr</i> [2]	Индекс	<i>huff_sampr</i> [2]	Индекс	<i>huff_sampr</i> [2]
-28	01011000101	-8	0100	12	0101110
-24	010110000	-4	00	16	01011010
-20	010110011	0	1	20	010110010
-16	01011011	4	011	24	0101100011
-12	0101111	8	01010	28	01011000100

Таблица Б.8 - *huff_sampr* [3]

Индекс	<i>huff_sampr</i> [3]	Индекс	<i>huff_sampr</i> [3]	Индекс	<i>huff_sampr</i> [3]
-32	00010101	-8	01	16	00011
-24	000100	0	1	24	0001011
-16	0000	8	001	32	00010100

Таблица Б.9 - *huff_sfreqba*

Индекс	<i>huff_sfreqba</i>	Индекс	<i>huff_sfreqba</i>
0	101111110100101100100	1944	110001110
8	101111110100101100101	1952	111111110
16	101111110100101100110	1960	10000101
24	101111110100101100111	1968	00101110
32	101111110100101101000	1976	10000110
40	101111110100101101001	1984	00100011
48	101111110100101101010	1992	11100100
56	101111110100101101011	2000	10011010
64	101111110100101101100	2008	00101111
72	101111110100101101101	2016	111101100
80	101111110100101101110	2024	01111101

Продолжение таблицы Б.9			
Индекс	<i>huff_sfreqba</i>	Индекс	<i>huff_sfreqba</i>
88	101111110100101101111	2032	01110100
96	101111110100101110000	2040	111011010
104	101111110100101110001	2048	10111011
112	101111110100101110010	2056	10011011
120	101111110100101110011	2064	10010000
128	101111110100101110100	2072	00110000
136	10111111010011	2080	10111100
144	101111110100101110101	2088	01011011
152	101111110101	2096	111011011
160	10111111011	2104	01001011
168	100010000	2112	10111101
176	001010010	2120	10011100
184	01000010	2128	00110001
192	101111111	2136	110001111
200	111001101	2144	01111110
208	00010001	2152	10010001
216	111001110	2160	10110011
224	111100001	2168	111011100
232	101010101	2176	00001110
240	010000011	2184	10010010
248	1111101001	2192	10111110
256	111100010	2200	10100110
264	111001111	2208	00110010
272	100111011	2216	00001111
280	010111011	2224	00011000
288	1111101010	2232	110101111
296	100010001	2240	110110000
304	0111000011	2248	111111111
312	001010011	2256	11111001
320	1111101011	2264	00111100
328	000100100	2272	00010000
336	000100101	2280	10100111
344	1111101100	2288	01101101
352	000100110	2296	01011100
360	110000000	2304	00000000
368	00110110	2312	00111101
376	010000110	2320	01001100
384	011101010	2328	00011001
392	011101011	2336	00100100
400	00010100	2344	110110001
408	010111100	2352	00111110
416	111100011	2360	00100101
424	110000001	2368	11001000
432	010111101	2376	00100110
440	011101100	2384	00000001
448	011101101	2392	100011101
456	1101001100	2400	110010010
464	1010101001	2408	00110011
472	100111100	2416	10000111
480	00011111	2424	011001011
488	010111110	2432	01111111
496	1111101101	2440	00000010

Продолжение таблицы Б.9			
Индекс	<i>huff_sfreqba</i>	Индекс	<i>huff_sfreqba</i>
504	111010000	2448	101101000
512	110000010	2456	01001101
520	00000100	2464	111101101
528	00100000	2472	00100111
536	111010001	2480	101101001
544	111100100	2488	110010011
552	111010010	2496	111101110
560	110000011	2504	111000111
568	01000100	2512	101101010
576	10010101	2520	00111111
584	01010001	2528	100100110
592	10101001	2536	111011101
600	101010110	2544	010011100
608	111100101	2552	110110010
616	011101110	2560	101101011
624	00000101	2568	101001001
632	10001001	2576	01000000
640	01100000	2584	101101100
648	01100001	2592	011110011
656	101010111	2600	100000000
664	100111101	2608	110010100
672	111010011	2616	00011010
680	101011000	2624	01001111
688	10101101	2632	100100111
696	10000010	2640	111001010
704	110100111	2648	100000001
712	00000110	2656	100000010
720	100010100	2664	101101101
728	001010100	2672	110010101
736	110101000	2680	000101111
744	010111111	2688	001101000
752	01010010	2696	111001011
760	011000100	2704	1110011001
768	011101111	2712	1101100110
776	111100110	2720	011011100
784	110101001	2728	000110110
792	111110111	2736	111011110
800	110000100	2744	110010110
808	111010100	2752	111101111
816	011000101	2760	010011101
824	1010110010	2768	0001111011
832	111100111	2776	110010111
840	01110001	2784	010100000
848	11001110	2792	101101110
856	01110010	2800	101010000
864	011110000	2808	1110111110
872	110111001	2816	101010001
880	110101010	2824	100000011
888	101011100	2832	1110111111
896	00101011	2840	010100001
904	01000101	2848	100101000
912	11001111	2856	1111000000

Продолжение таблицы Б.9			
Индекс	<i>huff_sfreqba</i>	Индекс	<i>huff_sfreqba</i>
920	01100011	2864	110011000
928	110111100	2872	110110100
936	011001000	2880	1010110011
944	011001001	2888	1101100111
952	011001010	2896	1111110001
960	110111101	2904	001101001
968	101011101	2912	1101101010
976	000100111	2920	11011010110
984	00110111	2928	110011001
992	01100110	2936	00101000
1000	10000011	2944	11011010111
1008	00111000	2952	011011101
1016	011110001	2960	001101010
1024	110000101	2968	1001010010
1032	101011110	2976	1101101100
1040	1111110000	2984	011011110
1048	100111110	2992	00110101100
1056	10111000	3000	000110111
1064	10111001	3008	101101111
1072	11010110	3016	1001010011
1080	11011101	3024	0101110101
1088	111010101	3032	110011010
1096	100111111	3040	1111000001
1104	101000000	3048	0000001100
1112	101011111	3056	1111101000
1120	110000110	3064	1001110100
1128	110111110	3072	01110000101
1136	00000111	3080	0110111110
1144	101000001	3088	110011011
1152	010000111	3096	00110101101
1160	101100000	3104	11011011010
1168	001010101	3112	1101101110
1176	110111111	3120	1011111100
1184	00111001	3128	0000001101
1192	111111001	3136	0110111111
1200	110000111	3144	000111100
1208	101000010	3152	10011101010
1216	111000000	3160	11011011011
1224	101100001	3168	11011011110
1232	101100010	3176	0011010111
1240	111000001	3184	0000001110
1248	111000010	3192	0111000000
1256	01010011	3200	11011011111
1264	00111010	3208	01000001000
1272	111000011	3216	101111110100101110110
1280	1101001101	3224	110111000000
1288	111101000	3232	1101110001
1296	110101011	3240	10011101011
1304	111000100	3248	110111000001
1312	01010100	3256	1110011000
1320	111111010	3264	0000001111
1328	11010000	3272	0111000001

Продолжение таблицы Б.9			
Индекс	<i>huff_sfreqba</i>	Индекс	<i>huff_sfreqba</i>
1336	01100111	3280	101111110100101110111
1344	01101000	3288	110111000010
1352	101000011	3296	010000010010
1360	00001000	3304	01000001010
1368	11000100	3312	010000010011
1376	10001011	3320	10101010000
1384	011110010	3328	0100000101100
1392	111101001	3336	0001111010
1400	10001100	3344	010000010111
1408	111101010	3352	101111110100101111000
1416	01010101	3360	0100000101101
1424	111000101	3368	110111000011
1432	00001001	3376	10101010001
1440	111010110	3384	01011101000
1448	111111011	3392	010111010010
1456	10100010	3400	010111010011
1464	11111000	3408	1011111101000
1472	01101001	3416	101111110100101111001
1480	01010110	3424	011100001000
1488	01101010	3432	101111110100101111010
1496	10001101	3440	101111110100101111011
1504	01101011	3448	101111110100101111100
1512	110101110	3456	101111110100101111101
1520	10010110	3464	101111110100101111110
1528	10010111	3472	101111110100101111111
1536	01010111	3480	011100001001
1544	01000110	3488	10111111010010000000
1552	111000110	3496	10111111010010000001
1560	01101100	3504	10111111010010000010
1568	00001010	3512	10111111010010000011
1576	01111010	3520	10111111010010000100
1584	00101100	3528	10111111010010000101
1592	111101011	3536	10111111010010000110
1600	00111011	3544	10111111010010000111
1608	01110011	3552	10111111010010001000
1616	01111011	3560	10111111010010001001
1624	11010001	3568	10111111010010001010
1632	111111100	3576	10111111010010001011
1640	100010101	3584	10111111010010001100
1648	01011000	3592	10111111010010001101
1656	10011000	3600	10111111010010001110
1664	100011100	3608	10111111010010001111
1672	10100011	3616	10111111010010010000
1680	110001010	3624	10111111010010010001
1688	0001110	3632	10111111010010010010
1696	00010101	3640	10111111010010010011
1704	10110010	3648	10111111010010010100
1712	111010111	3656	10111111010010010101
1720	01011001	3664	10111111010010010110
1728	00100001	3672	10111111010010010111
1736	00001011	3680	10111111010010011000
1744	00100010	3688	10111111010010011001

Окончание таблицы Б.9			
Индекс	<i>huff_sfreqba</i>	Индекс	<i>huff_sfreqba</i>
1752	01000111	3696	10111111010010011010
1760	00001100	3704	10111111010010011011
1768	111011000	3712	10111111010010011100
1776	11000110	3720	10111111010010011101
1784	11010010	3728	10111111010010011110
1792	00010110	3736	10111111010010011111
1800	01001000	3744	10111111010010100000
1808	101001000	3752	10111111010010100001
1816	110001011	3760	10111111010010100010
1824	01001001	3768	10111111010010100011
1832	10111010	3776	10111111010010100100
1840	01111100	3784	10111111010010100101
1848	000101110	3792	10111111010010100110
1856	00001101	3800	10111111010010100111
1864	10000100	3808	10111111010010101000
1872	01011010	3816	10111111010010101001
1880	00101101	3824	10111111010010101010
1888	01001010	3832	10111111010010101011
1896	111011001	3840	10111111010010101100
1904	11111101	3848	10111111010010101101
1912	10001111	3856	10111111010010101110
1920	10011001	3864	10111111010010101111
1928	10100101	3872	10111111010010110000
1936	101100011	3880	10111111010010110001

Таблица Б.10 - *huff_sfreqbr*

Индекс	<i>huff_sfreqbr</i>	Индекс	<i>huff_sfreqbr</i>
0	0000001011	1944	11000101001010
8	101111	1952	000011011111
16	10101	1960	000011100000
24	11011	1968	010100101101
32	11001	1976	111100101011
40	10100	1984	110001010011
48	10001	1992	010100101110
56	01100	2000	000011100001
64	01101	2008	000011100010
72	01011	2016	010100101111
80	00111	2024	000011100011
88	111101	2032	1100010101000
96	111110	2040	0111000000000
104	111010	2048	0111000000001
112	111001	2056	000011100100
120	100111	2064	010010000100011
128	101100	2072	000011100101
136	011111	2080	011100000001
144	100100	2088	100001111010
152	010101	2096	0111000000100
160	010000	2104	1100010101001
168	001100	2112	000011100110
176	001001	2120	1100010101010
184	000010	2128	000011100111

Продолжение таблицы Б.10			
Индекс	<i>huff_sfreqbr</i>	Индекс	<i>huff_sfreqbr</i>
192	000001	2136	0111000000101
200	000111	2144	000011101000
208	1111110	2152	110001010010110
216	1111000	2160	000011101001
224	1100011	2168	01001000010001011100010
232	1110111	2176	110001010010111
240	1101000	2184	000011101010
248	1001100	2192	11000101010110
256	1011010	2200	11000101010111
264	1100001	2208	0111000000110
272	1011100	2216	0111000000111
280	1000000	2224	1100010101100
288	1000010	2232	0111001101000
296	1001101	2240	1100010101101
304	0111101	2248	11000101011100
312	1001010	2256	0111001101001
320	0100010	2264	0111001101010
328	0111010	2272	000011101011
336	0100011	2280	11000101011101
344	0001011	2288	11000101011110
352	0100101	2296	01001000010001011100011
360	0100110	2304	11000101011111
368	0001100	2312	0111001101011
376	0010100	2320	110001011000000
384	0011010	2328	110001011000001
392	0010000	2336	1100010110001
400	0011011	2344	1100010110010
408	0010101	2352	110001011000010
416	11111110	2360	01001000010001011100100
424	0001000	2368	110001011000011
432	11101101	2376	11000101100110
440	11100000	2384	110001011001110
448	11010101	2392	01001000010001011100101
456	11100010	2400	01001000010001011100110
464	11010110	2408	11000101101000
472	11000001	2416	1000011110110
480	11101100	2424	11000101101001
488	10110110	2432	110001011001111
496	10111011	2440	11000101101010
504	10110111	2448	110001011010110
512	01001001	2456	1100010110110
520	00010011	2464	110001011010111
528	11010010	2472	110001011011100
536	01010011	2480	110001011011101
544	10000110	2488	110001011011110
552	01110110	2496	11000101110000
560	00101110	2504	01001000010001011100111
568	01001110	2512	110001011011111
576	01010000	2520	01001000010001011101000
584	01001111	2528	11000101110001
592	00101101	2536	01001000010001011101001
600	00000011	2544	01001000010001011101010
608	00000001	2552	110001011100100

Продолжение таблицы Б.10			
Индекс	<i>huff_sfreqbr</i>	Индекс	<i>huff_sfreqbr</i>
616	111100111	2560	01001000010001011101011
624	111100110	2568	110001011100101
632	110100110	2576	01001000010001011101100
640	111100100	2584	01001000010001011101101
648	00001111	2592	110001011100110
656	00101111	2600	110001011100111
664	111111110	2608	1100010111010
672	101110101	2616	01001000010001011101110
680	110101110	2624	110001011101100
688	100101100	2632	01001000010001011101111
696	00010100	2640	110001011101101
704	110101111	2648	01001000010001011110000
712	111000011	2656	01001000010001011110001
720	011100011	2664	01001000010001011110010
728	110100111	2672	110001011101110
736	100000100	2680	110001011101111
744	100000101	2688	110001011110000
752	011101111	2696	110001011110001
760	011100001	2704	110001011110010
768	100001110	2712	11000101111010
776	101110100	2720	11000101111011
784	000110110	2728	110001011110011
792	011100100	2736	110001011111000
800	000100101	2744	01001000010001011110011
808	001011001	2752	01001000010001011110100
816	1111111110	2760	01001000010001011110101
824	1111001011	2768	110001011111101
832	100101101	2776	01001000010001011110110
840	010010001	2784	1100010111111001
848	001000101	2792	01001000010001011110111
856	010100010	2800	01001000010001011111000
864	1100000001	2808	1100010111111100
872	011110010	2816	01001000010001011111001
880	1001011101	2824	1100010111111101
888	000110111	2832	01001000010001011111010
896	000101010	2840	110001011111111
904	1110001100	2848	01001000010001011111011
912	000000100	2856	01001000010001011111100
920	000011000	2864	111000111000000
928	1101010011	2872	010010000100010111111101
936	1110000100	2880	111000111000001
944	000100100	2888	111000111000010
952	011100010	2896	010010000100010111111110
960	1110001101	2904	01001000010001011111111
968	0111100000	2912	01001000010001000000000
976	000101011	2920	11100011100010
984	0111100001	2928	01001000010001000000001
992	1101010000	2936	111000111000011
1000	0111000001	2944	111000111000110
1008	1101010001	2952	01001000010001000000010
1016	0111001010	2960	111000111000111
1024	0111001011	2968	00000010100000
1032	1000011111	2976	01001000010001000000011

Продолжение таблицы Б.10			
Индекс	<i>huff_sfreqbr</i>	Индекс	<i>huff_sfreqbr</i>
1040	1100000010	2984	0100100001000100000100
1048	0010001001	2992	0100100001000100000101
1056	1101010010	3000	0100100001000100000110
1064	1100000000	3008	0100100001000100000111
1072	0111001110	3016	0100100001000100001000
1080	1110001111	3024	00000010100001
1088	1111111110	3032	0100100001000100001001
1096	0000111011	3040	0100100001000100001010
1104	0111100010	3048	0100100001000100001011
1112	0111100011	3056	0100100001000100001100
1120	0001101000	3064	0100100001000100001101
1128	0111001100	3072	00000010100010
1136	0111100111	3080	0100100001000100001110
1144	1001011110	3088	0100100001000100001111
1152	0010001100	3096	0100100001000100010000
1160	110000001100	3104	00000010100011
1168	0111001111	3112	0100100001000100010001
1176	1111111111	3120	01001000010000
1184	0111011100	3128	0100100001000100010010
1192	0010001101	3136	0100100001000100010011
1200	1000001100	3144	0100100001000100010100
1208	0010001110	3152	0100100001000100010101
1216	0010110001	3160	0100100001000100010110
1224	0111011101	3168	0100100001000100010111
1232	1001011110	3176	0100100001000100011000
1240	0100100000	3184	0100100001000100011001
1248	01111001101	3192	0100100001000100011010
1256	0000000000	3200	0100100001000100011011
1264	1001011100	3208	0100100001000100011100
1272	0001101001	3216	0100100001000100011101
1280	0000000001	3224	0100100001000100011110
1288	0000000010	3232	0100100001000100011111
1296	01001000011	3240	0100100001000100100000
1304	11100001010	3248	0100100001000100100001
1312	110000001101	3256	0100100001000100100010
1320	0001101010	3264	0100100001000100100011
1328	0000000011	3272	0100100001000100100100
1336	0001101011	3280	0100100001000100100101
1344	01110011011	3288	0100100001000100100110
1352	1001011111	3296	0100100001000100100111
1360	01010001100	3304	0100100001000100101000
1368	11100011101	3312	0100100001000100101001
1376	01010001101	3320	0100100001000100101010
1384	110000001110	3328	0100100001000100101011
1392	100001111011	3336	0100100001000100101100
1400	11100001011	3344	0100100001000100101101
1408	11000100000	3352	0100100001000100101110
1416	01111001100	3360	0100100001000100101111
1424	00100011110	3368	0100100001000100110000
1432	0010001000	3376	0100100001000100110001
1440	11000100001	3384	0100100001000100110010
1448	11000100010	3392	0100100001000100110011
1456	01010001110	3400	0100100001000100110100

Продолжение таблицы Б.10			
Индекс	<i>huff_sfreqbr</i>	Индекс	<i>huff_sfreqbr</i>
1464	01010001111	3408	0100100001000100110101
1472	0000110010	3416	0100100001000100110110
1480	111000111001	3424	0100100001000100110111
1488	00100011111	3432	0100100001000100111000
1496	10000011010	3440	0100100001000100111001
1504	11000100011	3448	0100100001000100111010
1512	00000010101	3456	0100100001000100111011
1520	10000011011	3464	0100100001000100111100
1528	111100101000	3472	0100100001000100111101
1536	10000011100	3480	0100100001000100111110
1544	01010010000	3488	0100100001000100111111
1552	010010000101	3496	0100100001000101000000
1560	01010010001	3504	0100100001000101000001
1568	110000001111	3512	0100100001000101000010
1576	00001100110	3520	0100100001000101000011
1584	0100100001001	3528	01001000010001010000100
1592	0101001001000	3536	01001000010001010000101
1600	110001001000	3544	01001000010001010000110
1608	00001100111	3552	01001000010001010000111
1616	100000111010	3560	0100100001000101001000
1624	1100010010010	3568	0100100001000101001001
1632	110001001010	3576	0100100001000101001010
1640	00101100000	3584	0100100001000101001011
1648	010100100101	3592	0100100001000101001100
1656	00001101000	3600	0100100001000101001101
1664	100000111011	3608	0100100001000101001110
1672	01010010011	3616	0100100001000101001111
1680	10000011110	3624	0100100001000101010000
1688	00101100001	3632	0100100001000101010001
1696	00001101001	3640	0100100001000101010010
1704	110001001011	3648	0100100001000101010011
1712	11000100110	3656	0100100001000101010100
1720	110001001110	3664	0100100001000101010101
1728	110001001111	3672	0100100001000101010110
1736	111100101001	3680	0100100001000101010111
1744	100000111110	3688	0100100001000101011000
1752	00001101010	3696	0100100001000101011001
1760	1100010010011	3704	0100100001000101011010
1768	00001101011	3712	0100100001000101011011
1776	00001101100	3720	0100100001000101011100
1784	00001101101	3728	0100100001000101011101
1792	110001010000	3736	0100100001000101011110
1800	100000111111	3744	0100100001000101011111
1808	000000101001	3752	0100100001000101100000
1816	1100010100010	3760	0100100001000101100001
1824	100001111000	3768	0100100001000101100010
1832	0101001001001	3776	0100100001000101100011
1840	010100101000	3784	0100100001000101100100
1848	111100101010	3792	0100100001000101100101
1856	000011011100	3800	0100100001000101100110
1864	010100101001	3808	0100100001000101100111
1872	0101001010100	3816	0100100001000101101000
1880	100001111001	3824	0100100001000101101001

Окончание таблицы Б.10			
Индекс	<i>huff_sfreqbr</i>	Индекс	<i>huff_sfreqbr</i>
1888	1100010100011	3832	0100100001000101101010
1896	010100101011	3840	0100100001000101101011
1904	000011011101	3848	0100100001000101101100
1912	1100010100100	3856	0100100001000101101101
1920	010100101100	3864	0100100001000101101110
1928	0101001010101	3872	0100100001000101101111
1936	000011011110	3880	0100100001000101110000

Таблица Б.11 - *huff_sfreqc*

Индекс	<i>huff_sfreqc</i>	Индекс	<i>huff_sfreqc</i>
0	0010001101110011010010000	1944	011101111
8	0010001101110011010010001	1952	11101110
16	0010001101110011010010010	1960	10011100
24	0010001101110011010010011	1968	11010011
32	0010001101110011010010100	1976	00011011
40	0010001101110011010010101	1984	110010111
48	0010001101110011010010110	1992	111100001
56	0010001101110011010010111	2000	11001110
64	0010001101110011010011000	2008	10011001
72	0010001101110011010011001	2016	111111100
80	0010001101110011011	2024	00001100
88	0010001101110011010011010	2032	111010100
96	001000110111001100	2040	01000101
104	0010001101110010	2048	11110000
112	00100011011100111	2056	0100111
120	100110101011010	2064	00001110
128	100110101011000	2072	01000010
136	100110101011001	2080	111111101
144	0010001101100	2088	111101100
152	0010001101111	2096	00011100
160	011001001001	2104	01110101
168	111101000111	2112	01100000
176	11000000000	2120	10001111
184	0111000010	2128	10111110
192	0010011010	2136	10010101
200	11100010010	2144	1010010
208	0000101111	2152	11100100
216	0000001100	2160	01010111
224	10110110111	2168	00101010
232	11101011000	2176	01110011
240	11011100010	2184	10101011
248	10111101000	2192	11011000
256	10101001000	2200	01111101
264	01101011101	2208	00101011
272	10001011101	2216	00110101
280	11001101101	2224	11000011
288	0101100011	2232	10001001
296	001011100	2240	111100110
304	0000001000	2248	111100010
312	11000000001	2256	10010100
320	01100100101	2264	11010111
328	10011010001	2272	11011101

Продолжение таблицы Б.11			
Индекс	<i>huff_sfreq</i>	Индекс	<i>huff_sfreq</i>
336	0001100010	2280	11111111
344	001010000	2288	10111000
352	110001001	2296	10110111
360	01101011100	2304	11000111
368	10101110100	2312	01100110
376	0000000001	2320	11010101
384	1000001101	2328	10110011
392	0111000011	2336	111101010
400	0011101010	2344	110111100
408	000110010	2352	01001010
416	1111011110	2360	10000000
424	11101101001	2368	10101110
432	01101000000	2376	0001001
440	10001010001	2384	01010101
448	111101000110	2392	01010011
456	00100011010	2400	01111001
464	11011100000	2408	10010110
472	0000001101	2416	10101000
480	1001001011	2424	01011101
488	11100111000	2432	11010001
496	01101000001	2440	11001100
504	11101011001	2448	00100000
512	1011001011	2456	10111010
520	1101010011	2464	0001111
528	1100010000	2472	11101000
536	0000100010	2480	00101111
544	0000101110	2488	01110110
552	0110101101	2496	11100101
560	1001110111	2504	10101010
568	001110100	2512	10011111
576	011100011	2520	11111011
584	001000111	2528	10001101
592	110000011	2536	10000111
600	111111001	2544	11101100
608	10110110110	2552	0100000
616	11110100010	2560	01000011
624	1000001100	2568	111101011
632	011010010	2576	111101001
640	100001010	2584	00111101
648	100100110	2592	10101111
656	1010001011	2600	0010110
664	0000100011	2608	11101001
672	11001111111	2616	10011000
680	1100000001	2624	11001001
688	01011100	2632	10110100
696	00000111	2640	01010001
704	110001101	2648	10100110
712	0100100111	2656	10100011
720	1010100101	2664	00110110
728	001100011	2672	10001100
736	0100100110	2680	01111111
744	0101000010	2688	01001011
752	010100000	2696	0111101

Продолжение таблицы Б.11			
Индекс	<i>huff_sfreq</i>	Индекс	<i>huff_sfreq</i>
760	000110000	2704	11010000
768	011111100	2712	101111011
776	110000001	2720	110110101
784	00110111	2728	01011010
792	00000100	2736	10001000
800	11001111110	2744	11000010
808	10001010000	2752	11100001
816	11011100001	2760	10100000
824	11100010011	2768	01011001
832	111100011	2776	111110011
840	100001011	2784	10010111
848	111000110	2792	00111000
856	1011101111	2800	01011110
864	11010100100	2808	000010110
872	0110101100	2816	110010100
880	0110010011	2824	11100110
888	1001001010	2832	01100111
896	1111011111	2840	111010111
904	100100000	2848	00001101
912	00111100	2856	111110100
920	1100111110	2864	110101000
928	1000101111	2872	110101101
936	1011001010	2880	11100000
944	00010100	2888	111101110
952	1111011010	2896	00010101
960	10100010100	2904	00101001
968	10101110101	2912	011011111
976	11001101000	2920	011100000
984	0001100011	2928	10000001
992	000110011	2936	10111111
1000	010110000	2944	000000101
1008	100000111	2952	101111000
1016	010101000	2960	01100011
1024	0110101111	2968	111110010
1032	10111101001	2976	111010101
1040	11101101000	2984	10100001
1048	11001101100	2992	101101100
1056	0111000100	3000	01000100
1064	011101110	3008	110010000
1072	00110011	3016	100010110
1080	01111000	3024	00100001
1088	1001110110	3032	00111111
1096	11011100011	3040	01011011
1104	0101000011	3048	011111101
1112	1011110101	3056	01101110
1120	1111110000	3064	01010110
1128	1101100111	3072	10111001
1136	1110001000	3080	111000111
1144	011100100	3088	10110000
1152	01001101	3096	110011110
1160	011001000	3104	111110101
1168	0111001010	3112	111110000
1176	1000101001	3120	00100111

Продолжение таблицы Б.11			
Индекс	<i>huff_sfreq</i>	Индекс	<i>huff_sfreq</i>
1184	110110010	3128	01001000
1192	10110001	3136	011000100
1200	0111001011	3144	101111001
1208	10011010000	3152	00100010
1216	00111010111	3160	01110100
1224	10100010101	3168	110010101
1232	11100111001	3176	101001111
1240	1100100011	3184	00100100
1248	101001110	3192	01100101
1256	100111010	3200	101101011
1264	111100111	3208	101010011
1272	001011101	3216	110101100
1280	1100010001	3224	001100010
1288	1111011011	3232	000010100
1296	1010111001	3240	1110101101
1304	0011001010	3248	1001101001
1312	0111000101	3256	1101100110
1320	001001100	3264	0010001100
1328	011011110	3272	1100100010
1336	11010010	3280	0000001001
1344	00001111	3288	000000111
1352	1010111000	3296	10101001001
1360	000110100	3304	10011010100
1368	010101001	3312	0110100001
1376	110111111	3320	00000110
1384	01000110	3328	0000101011
1392	1001101011	3336	1011101110
1400	1110011101	3344	0000000000
1408	010010010	3352	0011001011
1416	011010001	3360	11010100101
1424	101000100	3368	10001011100
1432	001010001	3376	1100110111
1440	011000101	3384	1011011010
1448	101110110	3392	11001101001
1456	10011110	3400	00001010101
1464	10010001	3408	00001010100
1472	000010000	3416	0010001101101
1480	1010111011	3424	0101100010
1488	1100110101	3432	0010011011
1496	1111110001	3440	011001001000
1504	010111111	3448	001000110111000
1512	1111010000	3456	10011010101110
1520	111100101	3464	10011010101111
1528	111100100	3472	1001101010101
1536	0001011	3480	00111010110
1544	01001100	3488	001000110111010
1552	000110101	3496	1001101010100
1560	1110110101	3504	0010001101110011010011011
1568	010111110	3512	0010001101110011010011100
1576	101100100	3520	1001101010110111
1584	110010110	3528	001000110111011
1592	100100111	3536	0010001101110011010011101
1600	111011011	3544	1001101010110110

Окончание таблицы Б.11			
Индекс	<i>huff_sfreq</i>	Индекс	<i>huff_sfreq</i>
1608	00001001	3552	00100011011100110100111110
1616	01101010	3560	00100011011100110100111111
1624	11111101	3568	00100011011100110101000000
1632	111001111	3576	0010001101110011010100001
1640	000000001	3584	0010001101110011010100010
1648	00111001	3592	0010001101110011010100011
1656	00100101	3600	0010001101110011010100100
1664	11000101	3608	0010001101110011010100101
1672	10001110	3616	0010001101110011010100110
1680	001100100	3624	0010001101110011010100111
1688	110110100	3632	0010001101110011010101000
1696	01100001	3640	0010001101110011010101001
1704	111011110	3648	0010001101110011010101010
1712	100100100	3656	0010001101110011010101011
1720	101101010	3664	0010001101110011010101100
1728	111000101	3672	0010001101110011010101101
1736	00011101	3680	0010001101110011010101110
1744	01111100	3688	0010001101110011010101111
1752	10000100	3696	0010001101110011010110000
1760	00000001	3704	0010001101110011010110001
1768	00000101	3712	0010001101110011010110010
1776	111011111	3720	0010001101110011010110011
1784	00111011	3728	0010001101110011010110100
1792	110000010	3736	0010001101110011010110101
1800	110001100	3744	0010001101110011010110110
1808	100010101	3752	0010001101110011010110111
1816	00010000	3760	0010001101110011010111000
1824	00010001	3768	0010001101110011010111001
1832	0110110	3776	0010001101110011010111010
1840	10000110	3784	0010001101110011010111011
1848	100100001	3792	0010001101110011010111100
1856	011010011	3800	0010001101110011010111101
1864	01000111	3808	0010001101110011010111110
1872	00110000	3816	0010001101110011010111111
1880	10000010	3824	001000110111001101000000
1888	00110100	3832	0010001101110011010000001
1896	110111101	3840	0010001101110011010000010
1904	110111001	3848	0010001101110011010000011
1912	01010010	3856	0010001101110011010000100
1920	10011011	3864	0010001101110011010000101
1928	11011011	3872	0010001101110011010000110
1936	110111110	3880	0010001101110011010000111

Таблица Б.12 - *huff_nlag*

Индекс	<i>huff_nlag</i>	Индекс	<i>huff_nlag</i>
-512	100010101101101100	4	110
-508	100010101101101101	8	1111
-504	100010101101101110	12	10010
-500	100010101101101111	16	100001
-496	100010111010000000	20	1000100
-492	100010111010000001	24	10000000
-488	100010111010000010	28	100000010

Продолжение таблицы Б.12			
Индекс	<i>huff_nlag</i>	Индекс	<i>huff_nlag</i>
-484	100010111010000011	32	1000101000
-480	100010111010000100	36	10001010011
-476	100010111010000101	40	100010111001
-472	100010111010000110	44	100010101110
-468	100010111010000111	48	1000101011010
-464	100010111010001000	52	1000101011110
-460	100010111010001001	56	10001010110111
-456	100010111010001010	60	1000101001000
-452	100010111010001011	64	10001010110011
-448	100010111010001100	68	100010101111100
-444	100010111010001101	72	10001010010010
-440	100010111010001110	76	100010111011110
-436	100010111010001111	80	100010100101110
-432	100010111010010000	84	10001011101110110
-428	100010111010010001	88	100010101111101
-424	100010111010010010	92	100010100101111
-420	100010111010010011	96	1000101110111110
-416	100010111010010100	100	10001011101110111
-412	100010111010010101	104	1000101110111111
-408	100010111010010110	108	1000101011011000
-404	100010111010010111	112	10001011101110011
-400	100010111010011000	116	1000101011011001
-396	100010111010011001	120	10000001100000000
-392	100010111010011010	124	1000101011011010
-388	100010111010011011	128	10000001100000001
-384	100010111010011100	132	10000001100000010
-380	100010111010011101	136	10000001100000011
-376	100010111010011110	140	10000001100000100
-372	100010111010011111	144	10000001100000101
-368	100010111010100000	148	10000001100000110
-364	100010111010100001	152	10000001100000111
-360	100010111010100010	156	10000001100001000
-356	100010111010100011	160	10000001100001001
-352	100010111010100100	164	1000000110000101
-348	100010111010100101	168	10000001100001100
-344	100010111010100110	172	10000001100001101
-340	100010111010100111	176	1000000110000111
-336	100010111010101000	180	10000001100010000
-332	100010111010101001	184	10000001100010001
-328	100010111010101010	188	10000001100010010
-324	100010111010101011	192	1000000110001010
-320	100010111010101100	196	10000001100010011
-316	100010111010101101	200	10000001100010110
-312	100010111010101110	204	10000001100010111
-308	100010111010101111	208	10000001100011000
-304	100010111010110000	212	10000001100011001
-300	100010111010110001	216	10000001100011010
-296	100010111010110010	220	1000000110001110
-292	100010111010110011	224	10000001100011011
-288	100010111010110100	228	10000001100011110
-284	100010111010110101	232	10000001100011111
-280	100010111010110110	236	10000001100100000
-276	100010111010110111	240	10000001100100001

Продолжение таблицы Б.12			
Индекс	<i>huff_nlag</i>	Индекс	<i>huff_nlag</i>
-272	100010111010111000	244	10000001100100010
-268	100010111010111001	248	10000001100100011
-264	100010111010111010	252	10000001100100100
-260	100010111010111011	256	10000001100100101
-256	100010111010111100	260	10000001100100110
-252	100010111010111101	264	10000001100100111
-248	100010111010111110	268	10000001100101000
-244	100010111010111111	272	10000001100101001
-240	100010111011000000	276	10000001100101010
-236	100010111011000001	280	10000001100101011
-232	100010111011000010	284	10000001100101100
-228	100010111011000011	288	10000001100101101
-224	100010111011000100	292	10000001100101110
-220	100010111011000101	296	10000001100101111
-216	100010111011000110	300	10000001100110000
-212	100010111011000111	304	10000001100110001
-208	100010111011001000	308	10000001100110010
-204	100010111011001001	312	10000001100110011
-200	100010111011001010	316	10000001100110100
-196	100010111011001100	320	10000001100110101
-192	100010111011001101	324	10000001100110110
-188	100010111011001110	328	10000001100110111
-184	100010111011001111	332	10000001100111000
-180	100010111011010000	336	10000001100111001
-176	100010111011010001	340	10000001100111010
-172	100010111011010010	344	10000001100111011
-168	100010111011010011	348	10000001100111100
-164	100010111011010100	352	10000001100111101
-160	100010111011010101	356	10000001100111110
-156	1000101001011001	360	10000001100111111
-152	100010111011010101	364	10000001101000000
-148	100010111011011000	368	10000001101000001
-144	100010111011011001	372	10000001101000010
-140	100010111011011010	376	10000001101000011
-136	100010111011011011	380	10000001101000100
-132	1000101110110110110	384	10000001101000101
-128	1000101110110110111	388	10000001101000110
-124	100010111011100000	392	10000001101000111
-120	100010111011100001	396	10000001101001000
-116	100010111011100010	400	10000001101001001
-112	100010111011100011	404	10000001101001010
-108	100010111011100100	408	10000001101001011
-104	100010111011100101	412	10000001101001100
-100	100010111011100110	416	10000001101001101
-96	1000101001011010	420	10000001101001110
-92	1000101001001111	424	10000001101001111
-88	1000101001011011	428	10000001101010000
-84	1000101110111010	432	10000001101010001
-80	1000101001010100	436	10000001101010010
-76	1000101001010101	440	10000001101010011
-72	10001010110010	444	10000001101010100
-68	10000001101100	448	10000001101010101
-64	10000001101101	452	10000001101010110

Окончание таблицы Б.12			
Индекс	<i>huff_nlag</i>	Индекс	<i>huff_nlag</i>
-60	10001010111111	456	1000000110101011
-56	1000000110111	460	10000001101011000
-52	1000101011000	464	10000001101011001
-48	100010101010	468	10000001101011010
-44	100010101011	472	10000001101011011
-40	100010111000	476	10000001101011100
-36	10001010100	480	10000001101011101
-32	1000000111	484	10000001101011110
-28	1000101111	488	10000001101011111
-24	100010110	492	10001010010011000
-20	1000001	496	10001010010011001
-16	100011	500	10001010010011010
-12	10011	504	10001010010011011
-8	1110	508	10001010010110000
-4	101	512	10001010010110001
0	0		

Таблица Б.13 - *huff_nlsf*

Индекс	<i>huff_nlsf</i>	Индекс	<i>huff_nlsf</i>	Индекс	<i>huff_nlsf</i>
7	10110011	14	1000	21	1011010
8	101101110	15	00	22	10010011
9	101101111	16	11	23	100100101
10	10110010	17	01	24	100100100
11	1001000	18	1010	25	10110110
12	1011000	19	10111		
13	100101	20	10011		

Таблица Б.14 - *huff_ngain*

Индекс	<i>huff_ngain</i>	Индекс	<i>huff_ngain</i>	Индекс	<i>huff_ngain</i>
-13	010011100	-4	0100010	5	01000111
-12	010011011001	-3	010010	6	010011001
-11	010011101110	-2	01011	7	0100110111
-10	010011101011	-1	00	8	0100110100
-9	01001110100	0	1	9	01001101101
-8	01001110110	1	011	10	010011101111
-7	0100110101	2	01010	11	010011101010
-6	010011000	3	010000	12	010011011000
-5	01000110	4	01001111		

Таблица Б.15 - *huff_scont*

Индекс	<i>huff_scont</i>	Индекс	<i>huff_scont</i>	Индекс	<i>huff_scont</i>
0	1000	4	001	8	0000
1	0100	5	1001	9	11
2	101	6	0101		
3	011	7	0001		

Таблица Б.16 - *huff_nrofbirths*

Индекс	<i>huff_nrofbirths</i>	Индекс	<i>huff_nrofbirths</i>	Индекс	<i>huff_nrofbirths</i>
0	010	21	000001001	42	000000100001110
1	1010	22	00000101	43	000000100001111
2	1110	23	000001100	44	000000100010000
3	011	24	00000001	45	000000100010001
4	110	25	0000010001	46	000000100010010
5	100	26	0000011010	47	000000100010011
6	001	27	0000011011	48	000000100010100
7	1111	28	000000100000100	49	000000100010101
8	0001	29	0000011100	50	000000100010110
9	10111	30	000000100000101	51	000000100010111
10	10110	31	000000100000110	52	000000100011000
11	0000111	32	000000100000111	53	000000100011001
12	000010	33	0000011101	54	000000100011010
13	0000110	34	000000100001000	55	000000100011011
14	00000011	35	0000011110	56	000000100011100
15	00000000	36	000000100001001	57	000000100011101
16	000000101	37	000000100001010	58	000000100011110
17	0000001001	38	0000011111	59	000000100011111
18	0000010000	39	000000100001011	60	000000100000000
19	000000100000010	40	000000100001100		
20	000000100000011	41	000000100001101		

Таблица Б.17 - *huff_ipd_df* и *huff_ipd_dt*

Индекс	<i>huff_ipd_df</i>	<i>huff_ipd_dt</i>
0	1	1
1	000	010
2	0110	0010
3	0100	00011
4	0010	00010
5	0011	0000
6	0101	0011
7	0111	011

Таблица Б.18 - *huff_opd_df* и *huff_opd_dt*

Индекс	<i>huff_opd_df</i>	<i>huff_opd_dt</i>
0	1	1
1	001	010
2	0110	0001
3	0100	00111
4	01111	00110
5	01110	0000
6	0101	0010
7	000	011

Таблица Б.19 - *huff_iid_df*[1] и *huff_iid_dt*[1]

Индекс	<i>huff_iid_df</i> [1]	<i>huff_iid_dt</i> [1]	Индекс	<i>huff_iid_df</i> [1]	<i>huff_iid_dt</i> [1]
-30	01111111010110100	0100111011010100	1	010	00
-29	01111111010110101	0100111011010101	2	0001	01011
-28	01111111010110110	0100111011001110	3	01101	010010
-27	01111111010110111	0100111011001111	4	011101	0100001
-26	01111111010110100	0100111011001100	5	0111101	01001100
-25	01111111010110101	0100111011010110	6	01111101	010011011
-24	011111111010001010	0100111011011000	7	011111100	0100111010
-23	011111111010001011	0100111101000110	8	0111111100	01001111001
-22	011111111010001000	0100111101100000	9	01111111100	01001110000
-21	01111111010000000	010011100011000	10	01111110100	010011101111
-20	011111111010110110	010011100011001	11	011111101011	010011100010
-19	011111111010000010	010011101100100	12	0111111101010	0100111101010
-18	01111111010111000	010011101100101	13	01111111101010	0100111011000
-17	0111111101000010	010011101101101	14	01111111010110	01001111010111
-16	0111111110101110	010011110110001	15	011111111010000	01001111010000
-15	011111110101111	01001110110111	16	0111111110101111	010011110110010
-14	01111111010001	01001111010110	17	01111111101000011	010011110100010
-13	01111111101001	0100111000111	18	011111111010111001	010011100011010
-12	0111111101001	0100111101001	19	011111111010000011	010011100011011
-11	011111101010	0100111101101	20	011111111010110111	0100111101100110
-10	011111111011	010011101110	21	011111111010000001	0100111101100111
-9	01111111011	010011110111	22	0111111111010001001	0100111101100001
-8	0111111011	01001111000	23	0111111111010001110	0100111101000111
-7	011111111	0100111001	24	0111111111010001111	0100111011011001
-6	01111100	010011010	25	011111111010001100	0100111011010111
-5	01111100	010011111	26	0111111111010001101	0100111011001101
-4	011100	0100000	27	0111111111010110010	0100111011010010
-3	01100	010001	28	0111111111010110011	0100111011010011
-2	0000	01010	29	0111111111010110000	0100111011010000
-1	001	011	30	0111111111010110001	0100111011010001
0	1	1			

Таблица Б.20 - *huff_icc_dt* и *huff_icc_df*

Индекс	<i>huff_icc_df</i>	<i>huff_icc_dt</i>
-7	11111111111111	11111111111110
-6	11111111111110	1111111111110
-5	111111111110	11111111110
-4	1111111110	111111110
-3	11111110	1111110
-2	11110	11110
-1	110	110
0	0	0
1	10	10
2	110	110
3	11110	11110
4	1111110	1111110
5	11111110	11111110
6	111111110	111111110
7	1111111110	11111111110

Таблица Б.21 - *huff_iid_df*[0] и *huff_iid_dt*[0]

Индекс	<i>huff_iid_df</i> [0]	<i>huff_iid_dt</i> [0]
-14	11111111111111011	11111111111111001
-13	11111111111111100	11111111111111010
-12	11111111111111101	11111111111111011
-11	11111111111111010	11111111111111000
-10	11111111111111100	11111111111111001
-9	1111111111111100	11111111111111010
-8	1111111111101	111111111111101
-7	111111110	1111111111110
-6	11111110	1111111110
-5	111110	11111110
-4	111100	1111110
-3	11101	11110
-2	1101	110
-1	101	10
0	0	0
1	100	110
2	1100	1110
3	11100	11110
4	111101	1111110
5	111110	11111110
6	1111110	111111110
7	11111110	1111111110
8	1111111100	111111111100
9	11111111100	11111111111000
10	111111111101	111111111111011
11	1111111111101	1111111111111100
12	11111111111110	1111111111111101
13	111111111111110	1111111111111110
14	111111111111111	1111111111111111

Приложение В

(справочное)

Темп и подача, масштабируемые в декодере

Темп, масштабирующий аудиосигнал, необходим, чтобы изменить (мгновенную) продолжительность сигнала, сохраняя воспринятую подачу. Как правило, в темпе декодера масштабирование обрабатывается для каждого объекта отдельно:

Переходные процессы; масштабирование темпа обычно не применяется к переходному объекту. Перцепционно переходные процессы характеризуются временами атаки и временами затухания. Намного более естественный результат получается, когда продолжительность переходного процесса сохраняется.

Синусоиды; для синусоидального объектного темпа масштабирование может быть получено, применяя два изменения в декодере. Прежде всего длина окна синтеза должна быть изменена согласно масштабному коэффициенту времени, гарантирующему, что унитарное перекрытие добавляется. Во-вторых, новое фазовое значение получается для продолжений, применяя непрерывную фазовую функцию.

Шум; в пределах объекта шума темп может масштабироваться, масштабируя оба окна синтеза.

Стерео: масштабирование темпа делается, изменяя позицию параметра согласно масштабному коэффициенту времени.

Поскольку подача главным образом определяется синусоидальными частотами, один подход должен масштабировать только частоты синусоид $f_{new} = p \cdot f_{old}$ с масштабным коэффициентом подачи p .

В.1 Непрерывная фаза

Так как фазовые значения зависят от размера подкадра S , в случае темпа и масштабирования подачи декодер переключается на непрерывную фазу. Темп и масштабирующая подачу процедура для синусоидальных компонентов делается следующим образом. Во-первых, исходная частота и фаза получаются процессом декодирования. В случае *phase_jitter_present* устанавливается в %1, полученная частота повторно квантуется, приводя к уровню представления sf_n . Уровень представления частоты sf_n становится $sf_n + sf_{jitter}$ и деквантуется. Эта частота умножается на масштабный коэффициент подачи и используется в непрерывном фазовом вычислении.

Непрерывная фаза ϕ_2 вычисляется так, что фаза двух последовательных сегмен-

тов перекрывается в середине области.

Фазовая информация φ_1 и φ_2 определяется в течение середины сегмента ($= (L-1)/2$). Где f_1 , и f_2 представляют частоту в радианах.

Фаза продолжающегося синусоидального компонента в первом подкадре фрейма с *refresh_sinusoids* = %1 не устанавливается в значение, считанное из потока (*s_phi*), а вычисляется из выражения для φ_2 . Если процесс декодирования запускается в этом подкадре, наугад от отмеченного подкадра, фаза продолжающегося синусоидального компонента должна быть установлена в значение, считанное из потока.

В.2 Темп, масштабирующийся для стерео

Темп, масштабирующийся для параметрического инструмента стерео, может быть достигнут, адаптируя позицию параметров. μ - масштабный коэффициент темпа. Считывается предыдущее и текущее масштабируемыми позициями параметры $\hat{n}_{prev} = n_{e-1} \cdot \mu$ и $\hat{n}_{curr} = n_e \cdot \mu$ соответственно. Перемещая эти позиции в целочисленные позиции, декодирование стерео может быть выполнено как обычно. Метод, для отображения позиции параметра нецелого числа \hat{n}_{prev} и \hat{n}_{curr} к целочисленным позициям n_{prev} и n_{curr} дается, используя следующую рекурсию.

$$m = \text{mod}(\hat{n}_{curr} - n_{prev}, 1)$$

где n_{prev} является предыдущей целочисленной позицией.

Затем текущая целочисленная позиция параметра вычисляется следующим образом:

$$n_{curr} = \hat{n}_{curr} + 1 - m$$

Чтобы инициировать рекурсию $n_{prev} = 0$.

Библиография

- [1] ИСО/МЭК 14496-3:2009 Информационные технологии. Кодирование аудиовизуальных объектов. Часть 3. Аудио (ИСО/МЭК14496-3:2009 *Information technology - Coding of audio-visual objects - Part 3: Audio*)

УДК 621.396:006.354

ОКС 33.170

Ключевые слова: звуковое вещание, электрические параметры, каналы и тракты, технологии MPEG-кодирования, синтетический звук, масштабирование, защита от ошибок, поток битов расширения, психоакустическая модель

Подписано в печать 30.04.2014. Формат 60x84^{1/8}.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ФГУП «СТАНДАРТИНФОРМ»

123995 Москва, Гранатный пер., 4.

www.gostinfo.ru

info@gostinfo.ru